2171 Bos 509/

03-25-04

PATENT
Customer No. 22,852
Attorney Docket No. 7451.0007-02000
InterTrust Ref. No.: IT-11.2 (US)

CERTIFICATE OF EXPRESS MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service's "Express Mail Post Office to Addressee" service under 37 CFR § 1.10, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on March 23, 2004. Express Mail Label Nos.: **EV 398887719 US**

Signed: Cindy Baglietto

Cindy Baglietto

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	
HALL, Edwin J.	Group Art Unit: 2171
Application No.: 09/819,063	Examiner: Amsbury, Wayne P.
Filed: September 28, 2000	
For: TECHNIQUES FOR DEFINING, USING, AN D MANIPULATING RIGHTS MANAGEMENT DATA STRUCTURES	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

THIRD UPDATED NOTICE REGARDING RELATED LITIGATION

Further to the submission of the Second Updated Notice Regarding Related Litigation on July 31, 2003, Applicants submit this Third Updated Notice to inform the Examiner of the status of the ongoing litigation between InterTrust and Microsoft, captioned InterTrust Tech. Corp. v. Microsoft Corp. (C 01-1640 SBA, N. D. Ca.). Applicants also submit herewith potentially relevant copies of the papers exchanged by the parties in the course of this litigation.

STATUS OF RELATED LITIGATION

Dated September 2, 2003, and attached as Exhibit 1 hereto, is InterTrust's Disclosures of Asserted Claims and Preliminary Infringement Contentions Pursuant to Patent Local Rules 3-1 and 3-2 with Exhibits A and B. Exhibit C has not been provided because (1) it is marked "Confidential - Subject to Protective Order" and "Attorneys Eyes Only" (as it pertains to proprietary Microsoft information); and (2) it is not material to the patentability of the pending claims, as it contains only information pertaining to Microsoft's current products and systems.

On November 17, 2003, Microsoft filed Defendant Microsoft Corporation's Preliminary Invalidity Contentions (Patent Local Rules 3-3 and 3-4). *See* Exhibit 2.

REMARKS

Applicants submit this Third Updated Notice Regarding Related Litigation in fulfillment of their duty to disclose information potentially material to patentability under 37 CFR 1.56. Applicants encourage the Examiner to carefully review the attached documents, and let Applicants know if any additional information is desired.

With this Notice, Applicants have provided copies of the papers described in the Status of Related Litigation section above. Furthermore, a voluminous number of documents have been referred to in the Microsoft paper attached as Exhibit 2 (specifically, in Exhibit A, attached thereto). All of the references listed in Exhibit A which have not already been cited in this application are listed on an Information Disclosure Statement filed March 22, 2004, and copies of the cited documents were provided on CD-ROM therewith. Furthermore, Applicants urge the Examiner to also

review Exhibits B and C, exhibits to Microsoft's Preliminary Invalidity Contentions, which comprise an extensive listing of claim charts pertaining to the patents-in-suit. Exhibits B and C are provided in electronic format (via CD-ROM, sent with this Notice) due to their sizeable page length.

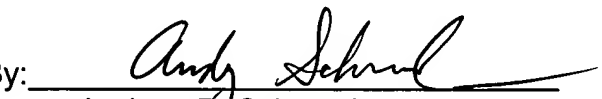
As always, if the Examiner believes that any document not yet submitted may be helpful in resolving an issue before him and would like to review that or any other document, Applicants invite the Examiner to contact the undersigned at (650) 849-6643.

If there are any fees due with the filing of this Notice which have not yet been paid, please charge the fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: March 23, 2004

By: 
Andrew B. Schwaab
Reg. No. 38,611

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.
1300 I Street, N.W.
Washington, D.C. 20005-3315

KEKER & VAN NEST, LLP
JOHN W. KEKER - #49092
MICHAEL H. PAGE - #154913
710 Sansome Street
San Francisco, CA 94111-1704
Telephone: (415) 391-5400
Facsimile: (415) 397-7188

INTERTRUST TECHNOLOGIES CORPORATION
DOUGLAS K. DERWIN - #111407
JEFFERY J. McDOW - #184727
4800 Patrick Henry Drive
Santa Clara, CA 95054
Telephone: (408) 855-0100
Facsimile: (408) 855-0144

PENNIE & EDMONDS LLP
MICHAEL J. LYONS - #202284
300 Hillview Avenue
Palo Alto, CA 94304
Telephone: (650) 493-4935
Facsimile: (650) 493-5556

Attorneys for Plaintiff and Counter-Defendant
INTERTRUST TECHNOLOGIES CORPORATION

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

INTERTRUST TECHNOLOGIES CORPORATION, a Delaware corporation,

Plaintiff,

v.

MICROSOFT CORPORATION, a Washington corporation,

Defendant.

AND COUNTER ACTION.

Case No. C 01-1640 SBA (MEJ)

Consolidated with C 02-0647 SBA

INTERTRUST'S DISCLOSURES OF ASSERTED CLAIMS AND PRELIMINARY INFRINGEMENT CONTENTIONS PURSUANT TO PATENT LOCAL RULES 3-1 and 3-2

('683, '193, '861, '721, '891, '900, '912, '019, '876, '181, and '402 Patents)

Pursuant to the Court's August 8, 2003 Order, Plaintiff InterTrust Technologies Corporation ("InterTrust") hereby submits its Disclosures of Asserted Claims and Preliminary Infringement Contentions under Patent Local Rules 3-1 and 3-2 ("PLR 3-1 & 3-2 Disclosures") to Defendant Microsoft Corporation ("Microsoft"). These PLR 3-1 & 3-2 Disclosures supercede all previous PLR 3-1 and PLR 3-2 disclosures served by InterTrust in this case.

PATENT LOCAL RULE 3-1: DISCLOSURE OF ASSERTED CLAIMS AND PRELIMINARY INFRINGEMENT CONTENTIONS

(a) Asserted claims

InterTrust currently contends that the Microsoft products identified herein infringe the claims of U.S. Patents Nos. 6,185,683 B1 ("the '683 patent"); 6,253,193 B1 ("the '193 patent"); 5,920,861 ("the '861 patent"); 6,157,721 ("the '721 patent"); 5,982,891 ("the '891 patent"); 5,892,900 ("the '900 patent"); 5,917,912 ("the '912 patent"); 5,915,019 ("the '019 patent"); 5,949,876 ("the '876 patent"); 6,112,181 ("the '181 patent"); and 6,389,402 B1 ("the '402 patent"), as identified in the attached claim charts. As discovery progresses, InterTrust may determine that additional Microsoft products infringe the asserted patents and/or that Microsoft infringes additional patent claims. InterTrust reserves the right to supplement and/or amend its disclosures and infringement contentions.

(b) Accused products

InterTrust contends that various Microsoft products infringe the patent claims identified in the claim charts attached hereto. Accused products are listed in Exhibit A hereto. Accused products are listed in Exhibit A hereto, which is intended to encompass past, present, and future product versions that include the accused features and/or functionality.

(c) Claim charts

InterTrust submits the attached claim charts based solely on information available to it to date. Discovery is ongoing, and additional information is likely to be produced during discovery. InterTrust therefore reserves the right to supplement and/or amend its infringement assertions as discovery proceeds.

InterTrust contends that Microsoft infringes at least the claims of the '683, '193, '861, '721, '891, '900, '912, '019, '876, '181, and '402 patents identified in the claim charts attached hereto as Exhibits B and C:[1]

(d) Literal infringement and the doctrine of equivalents

InterTrust contends that Microsoft infringes the claims of the '683, '193, '861, '721, '891, '900, '912, '019, '876, '181, and '402 patents as specified in Exhibits B and C both literally and under the doctrine of equivalents.

(e) Priority from earlier applications

InterTrust claims priority for the claims of the '891, '912, '683, '193, '019, '876, and '402 patents-in-suit dating to application No. 08/388,107, filed February 13, 1995. InterTrust claims priority for the claims of the '900 patent-in-suit dating to application No. 08/695,927, filed August 12, 1996. InterTrust does not claim priority for the claims of the '721, '861, and '181 patents-in-suit dating to any earlier application.

(f) Reliance on InterTrust's own products

InterTrust does not currently intend to rely on the assertion that its own Commerce and Rights System products practice at least some of the claimed inventions of the '683, '193, '861, '721, '891, '900, '912, '019, '876, '181, and '402 patents-in-suit to support its infringement assertions against Microsoft.

PATENT LOCAL RULE 3-2: DOCUMENT PRODUCTION ACCOMPANYING DISCLOSURE

(a) Documents re disclosure and/or offer of sale

InterTrust is not currently aware of such documents other than the documents that have previously been produced. See IT00017664-19168, IT00020866-21695, IT00021700-23578,

---

[1] Exhibit B contains claim charts based upon publicly available or non-confidential sources. Exhibit C contains additional claim charts referencing material designated as "Attorneys' Eyes Only" by Microsoft, and is served under separate caption. No other information contained in these disclosures is designated confidential by either party, and InterTrust does not object to dissemination of this document, other than Exhibit C, to persons not permitted to view confidential information in this case. For ease of reference, the claim charts attached hereto include all claims previously disclosed by InterTrust, as well as new claims. Numbering/lettering/bolding has been added to the text of each claim for convenience only, and is not intended to alter, expand, or interpret the meaning of those claims. In instances where infringement claims are illustrated by quotation or reference to Microsoft documents, those

IT00038608-43419.

(b) Documents re conception, reduction to practice, and/or design/development

InterTrust has produced nonprivileged documents concerning the conception, design, development, and reduction to practice of the inventions disclosed in the patents-in-suit. See, e.g., IT00000005-17261, IT00036207-38606, IT00041497-549. In addition, InterTrust has produced voluminous archives of source code created in the course of its business, some of which may constitute additional evidence of the conception, design, development, and reduction to practice of its patented inventions. InterTrust is not currently aware of any other such nonprivileged documents in its possession or control other than said source code and the source code and documents that have been produced.

(c) Prosecution history of patents-in-suit

The prosecution histories of the patents-in-suit have previously been produced. See, e.g., IT00062350-67643, IT00070342-72434, FH00107455 – 107731, FH00113539-118857, FH118866-121322.

Dated: September 2, 2003

KEKER & VAN NEST, LLP

By: _____
MICHAEL H. PAGE
Attorneys for Plaintiff and Counter-Defendant
INTERTRUST TECHNOLOGIES CORPORATION

---

references are intended to be exemplary only, and not limiting.

Microsoft Accused Products

Visual Studio .Net Enterprise Architect
Visual Studio .NET Enterprise Developer
Visual Studio .NET Professional
Visual Studio .Net
ASP.Net
.NET Framework SDK
.Net License Compiler

Office XP Standard
Office XP Professional
Office XP Professional with FrontPage
Office XP Developer
Windows XP Home Edition
Windows XP Professional
Access 2002
Excel 2002
FrontPage 2002
Outlook® 2002
PowerPoint ® 2002
Project 2002
Publisher ® 2002
Visio® 2002
Word 2002
Visio Enterprise Network Tools
Office 2000 SR-1
Project 2000 SR-1
Windows XP Embedded
Windows CE .NET
Windows CE for Automotive
Mobility and Wireless Solutions for business
Mobile Devices
Pocket PC
Microsoft Smartphone Platform
Microsoft XBOX
Windows ME
Digital Asset Server
Microsoft Reader
Windows Media Player
Windows Media Rights Manager SDK
Windows Media Device DRM technology
Microsoft Secure Audio Path technology

Microsoft System Management Server
Windows File Protection System
Microsoft ActiveX technology, including all Microsoft tools that support the Microsoft ActiveX licensing model

All products that contain the Microsoft Common Language Runtime (CLR), Microsoft Compact CLR, or Microsoft implemented .Net Common Language Infrastructure

Application Center
BizTalk Server
Commerce Server
Content Management Server
Exchange Server
Host Integration Server
Internet Security and Acceleration Server
Mobile Information Server
SharePoint Portal Server
SQL Server
Windows 2000 Server
.NET Enterprise Services
.NET Infrastructure and Services

Microsoft Installer SDK
All products that contain the Microsoft Installer Technology

Microsoft .Net MyServices
Windows Hardware Quality Labs Certification Services

Office 2003 and included applications

Server 2003, including Microsoft hosted RMS Services using Passport

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,892,900

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
155.	Products infringing: Any product using Microsoft Product Activation or Reader Activation feature.
A virtual distribution environment comprising	
(a) a first host processing environment comprising	computer running a Microsoft product containing the Product Activation feature, including Windows XP, Office XP, Visio 2002. Reader using its activation feature.
(1) a central processing unit;	CPU of computer
(2) main memory operatively connected to said central processing unit;	main memory of computer
(3) mass storage operatively connected to said central processing unit and said main memory;	hard disk or other mass storage contained in computer
(b) said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising:	Microsoft Product Activation software
(1) machine check programming which derives information from one or more aspects of said host processing environment,	Product Activation software generates hardware information relating to the host processing environment as part of the activation process
(2) one or more storage locations storing said information;	hardware information is stored in the computer's storage
(3) integrity programming which	
(i) causes said machine check programming to derive said information,	each time the Microsoft program starts up after initial activation, Product Activation checks the originally derived hardware information against current hardware
(ii) compares said information to information previously stored in said one or more storage locations, and	each time the Microsoft program starts up after initial activation, Product Activation checks the originally derived hardware information against current hardware
(iii) generates an indication based on the result of said comparison; and	Product Activation software indicates whether the test has passed or failed
(4) programming which takes one or more actions based on the state of said indication;	
(i) said one or more actions including at least temporarily halting further processing.	Product Activation software will allow system startup procedures to continue, if test succeeds, or discontinue startup and offer user opportunity to reactivate if the test fails

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,892,900

156.	Product Infringing: Any product using Microsoft Product Activation or Reader Activation feature.
A virtual distribution environment comprising	
(a) a first host processing environment comprising	computer running a Microsoft product containing the Product Activation feature, including Windows XP, Office XP, Visio 2002 and Reader
(1) a central processing unit;	CPU of computer
(2) main memory operatively connected to said central processing unit;	main memory of computer
(3) mass storage operatively connected to said central processing unit and said main memory;	hard disk or other mass storage contained in computer
(b) said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising:	Microsoft Product Activation software
(1) machine check programming which derives information from one or more aspects of said host processing environment,	Product Activation software generates hardware information relating to the host processing environment as part of the activation process
(2) one or more storage locations storing said information;	hardware information is stored in the computer's storage
(3) integrity programming which	
(i) causes said machine check programming to derive said information,	each time the Microsoft program starts up after initial activation, Product Activation checks the originally derived hardware information against current hardware
(ii) compares said information to information previously stored in said one or more storage locations, and	each time the Microsoft program starts up after initial activation, Product Activation checks the originally derived hardware information against current hardware
(iii) generates an indication based on the result of said comparison; and	Product Activation software indicates whether the test has passed or failed
(4) programming which takes one or more actions based on the state of said indication;	
(i) said one or more actions including at least temporarily disabling certain functions.	Product Activation may disable the underlying software from generating new files or running user applications if the test fails

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,892,900

157.	Product Infringing: Any product using Microsoft Product Activation or Reader Activation feature.
A virtual distribution environment comprising	
(a) a first host processing environment comprising	computer running a Microsoft product containing the Product Activation feature, including Windows XP, Office XP, Visio 2002 and Reader
(1) a central processing unit;	CPU of computer
(2) main memory operatively connected to said central processing unit;	main memory of computer
(3) mass storage operatively connected to said central processing unit and said main memory;	hard disk or other mass storage contained in computer
(b) said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising:	Microsoft Product Activation software
(1) machine check programming which derives information from one or more aspects of said host processing environment,	Product Activation software generates hash information relating to the host processing environment as part of the activation process
(2) one or more storage locations storing said information;	hardware information is stored in the computer's storage
(3) integrity programming which	
(i) causes said machine check programming to derive said information,	each time the Microsoft program starts up after initial activation, Product Activation checks the originally derived hardware information against current hardware
(ii) compares said information to information previously stored in said one or more storage locations, and	each time the Microsoft program starts up after initial activation, Product Activation checks the originally derived hardware information against current hardware
(iii) generates an indication based on the result of said comparison; and	Product Activation software indicates whether the test has passed or failed
(4) programming which takes one or more actions based on the state of said indication;	
(i) said one or more actions including displaying a message to the user.	Product Activation software displays a message to the user if the test fails

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,892,900

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
156.	Products infringing: Windows Media Player
A virtual distribution environment comprising	
a first host processing environment comprising	WMP with Individualized DRM client (referred to hereafter as the Individualized WMP) running on a client computer
a central processing unit	Client CPU
main memory operatively connected to said central processing unit	Client memory
mass storage operatively connected to said central processing unit and said main memory	Local disk drive
said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising:	Individualized WMP (I-WMP) stored on disk and loaded into main memory upon execution. I-WMP is tamper resistant.
machine check programming which derives information from one or more aspects of said host processing environment,	Individualization module is generated by the MS individualization service either when the un-individualized WMP tries to open licensed content that requires a security upgrade (aka, Individualization) or when the user requests an upgrade un-provoked. The individualization module is unique and signed and is bound to a unique hardware ID using the MS machine activation process.
one or more storage locations storing said information	The aforementioned unique feature are located in multiple places or storage locations
integrity programming which	
causes said machine check programming to derive said information,	The ID is regenerated by WMP/DRM client when first loading the Individualized DRM Client to access a piece of content requiring the security upgrade.
compares said information to information previously stored in said one or more storage locations, and	The program checks the new copy against the one to which the Individualized DRM client is bound.
generates an indication based on the result of said comparison; and	Program stores the result of this check.
programming which takes one or more actions based on the state of said indication	If these are not equal, the user is notified via a message stating that he/she must acquire a security upgrade (that is, the current security upgrade is invalid). If they are equal then processing of songs requiring Individualization continues.
said one or more actions including at least temporarily disabling certain functions.	Songs targeted to this Individualization module cannot be accessed until the upgrade is correct.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,892,900

157. A virtual distribution environment comprising	Infringing products include: Windows Media Player
a first host processing environment comprising	See 156
a central processing unit	See 156
main memory operatively connected to said central processing unit	See 156
mass storage operatively connected to said central processing unit and said main memory	See 156
said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising:	See 156
machine check programming which derives information from one or more aspects of said host processing environment,	See 156
one or more storage locations storing said information	See 156
integrity programming which causes said machine check programming to derive said information compares said information to information previously stored in said one or more storage locations, and	See 156
generates an indication based on the result of said comparison; and	See 156
programming which takes one or more actions based on the state of said indication	See 156
said one or more actions including displaying a message to the user.	If these are not equal, the user is notified via a message stating that he/she must acquire a security upgrade (that is, the current security upgrade is invalid).

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,892,900

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
157.	Infringing Product: Microsoft's Windows File Protection and System File Checker features, embodied in Microsoft's Windows 2000, Windows XP products, and Server 2003
A virtual distribution environment comprising	
(a) a first host processing environment comprising	computer running Microsoft Windows 2000 or Windows XP.
(1) a central processing unit;	CPU of computer
(2) main memory operatively connected to said central processing unit;	main memory of computer
(3) mass storage operatively connected to said central processing unit and said main memory;	hard disk or other mass storage contained in computer
(b) said mass storage storing tamper resistant software designed to be loaded into said main memory and executed by said central processing unit, said tamper resistant software comprising:	Windows File Protection process/service ("WFP") and System File Checker (SFC.exe) features of winlogon.exe. Winlogon.exe is treated as a "critical" service by the Windows operating system. Files supporting WFP (including winlogon.exe, sfc.exe, sfc.dll (2000 only), sfcfiles.dll (2000 only) and sfc_os.dll (XP only)) are "protected" files and are signed using a signature verified by a hidden key. In Windows 2000, WFP uses hidden functions within the sfc.dll library. Functions are imported by "ordinal" instead of "name."
(1) machine check programming which derives information from one or more aspects of said host processing environment,	Winlogon either directly or using another dll (XP) or using SFC.dll (2000) determines if changed file was protected, computes the hash of protected files and, if necessary, computes the hash of the file in the dll cache before using it to replace a file overwritten by an incorrect version of the file.
(2) one or more storage locations storing said information;	hardware information is stored in the computer's memory
(3) integrity programming which	
(i) causes said machine check programming to derive said information,	Windows notifies Winlogon when there has been a system directory change or a change in the dll cache.
(ii) compares said information to information previously stored in said one or more storage locations, and	Winlogon either directly or using another dll (XP) or using SFC.dll (2000) compares computed hash with hash in the hash database created from the Catalog file(s), and, if there is a difference, compares the hash of the file in the dll cache to the hash database created from

	the Catalog file(s) before using it to replace an overwritten file.
(iii) generates an indication based on the result of said comparison; and	An event is written to the Event Viewer if hashes do not agree.
(4) programming which takes one or more actions based on the state of said indication;	Depending on the circumstances, WFP displays several messages to the user, including prompting the user to contact the system administrator, and to insert a CD-ROM.
(i) said one or more actions including displaying a message to the user.	See above. Messages also constitute viewable Event Property pop-ups.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,917,912

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
6.	Product Infringing: XBox
A process comprising the following steps:	The process constitutes assembly and use of components making up an XBox game.
accessing a first record containing information directly or indirectly identifying one or more elements of a first component assembly,	The first record consists of the second file table on an XBox DVD. This table identifies the .xbe file which includes the game information.
at least one of said elements including at least some executable programming,	The xbe file includes executable programming.
at least one of said elements constituting a load module,	The xbe file is a load module.
said load module including executable programming and a header;	The xbe file includes a header.
at least a portion of said header is a public portion which is characterized by a relatively lower level of security protection; and	Most information the xbe header is not obfuscated.
at least a portion of said header is a private portion which is characterized, at least some of the time, by a level of security protection which is relatively higher than said relatively lower level of security protection,	The entry point address and the kernel image thunk address listed in the xbe header are obfuscated and therefore at a higher level of security protection.
using said information to identify and locate said one or more elements;	The second file table identifies the .xbe file, including where that file is located.
accessing said located one or more elements;	The .xbe file is accessed by the XBox.
securely assembling said one or more elements to form at least a portion of said first component assembly;	At runtime, the .xbe file is assembled with certain services of the operating system to form a component assembly. Security associated with this assembling process includes verifying signatures associated with portions of the .xbe file, and replacing obfuscated calls to operating system services with actual addresses. The assembly may also include patch files downloaded from a remote server.
executing at least some of said executable	Game play requires execution of the

programming; and	assembled programming.
checking said record for validity prior to performing said executing step.	The second file table is protected by a digital signature, and is not loaded/used unless the digital signature is verified against the file.
7. A process as in claim 6 in which:	
said relatively lower level of security protection comprises storing said public header portion in an unencrypted state; and	The header is protected by the techniques protecting the xbe such as signing and security descriptors, but it is not encrypted except as noted below.
said relatively higher level of security protection comprises storing said private header portion in an encrypted state.	The **entry point address** and the **kernel image thunk address** listed in the xbe header are obfuscated. The Xbox SDK's (XDK) image build uses a key value shared with the retail XBox to perform two XOR operations against the addresses

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,917,912

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
8.	Infringing products: Microsoft CLR or CCLR and .NET Framework SDK and products that include one or both of these.
A process comprising the following steps:	
(a) accessing a first record containing information directly or indirectly identifying one or more elements of a first component assembly,	The first record is either an assembly manifest, or a whole assembly; the elements are other assemblies that are referenced as external in the first record; the first component assembly is a .NET application domain.
(1) at least one of said elements including at least some executable programming,	Assembly contains executable programming.
(2) at least one of said elements constituting a load module,	This is an external assembly referenced in the first record.
(i) said load module including executable programming and a header;	Assemblies include executable programming, and the assembly manifest and CLS type metadata constitute a header.
(ii) said header including an execution space identifier identifying at least one aspect of an execution space required for use and/or execution of the load module associated with said header;	This feature is provided for in the .NET architecture through numerous mechanisms, for example, by demands for ZoneID permissions.
(iii) said execution space identifier provides the capability for distinguishing between execution spaces providing a higher level of security and execution spaces providing a lower level of security;	SecurityZone or other evidence provides this capability.
(b) using said information to identify and locate said one or more elements;	Manifest and type metadata information section is used to identify and locate files, code elements, resource elements, individual classes and methods.
(c) accessing said located one or more elements;	Step carried out by the CLR or CCLR loader.
(d) securely assembling said one or more elements to form at least a portion of said first component assembly;	CLR or CCLR carries out this step, including checking the integrity of the load module, checking the load module's permissions, placing the load module contents into an application domain, isolating it from malicious or badly behaved code, and from code that does not have the permission to call it.
(e) executing at least some of said executable programming; and	Step carried out by the CLR/CCLR and the CLR/CCLR host.

(f) checking said record for validity prior to performing said executing step.	The CLR/CCLR checks the authenticity and the integrity of the first .NET assembly.
9. A process as in claim 8 in which said execution space providing a higher level of security comprises a secure processing environment.	The CLR/CCLR constitutes a secure processing environment.
13. A process as in claim 8 further comprising:	
(a) comparing said execution space identifier against information identifying the execution space in which said executing step is to occur; and	In one example, the ZoneIdentityPermissionAttribute SecurityZone value demanded by control in the assembly manifest is compared against the SecurityZone attribute value corresponding to the calling method
(b) taking an action if said execution space identifier requires an execution space with a security level higher than that of the execution space in which said executing step is to occur.	CLR/CCLR will throw an exception and transfer control to an exception handler in the calling routine, or it will shut down the application if there is no such exception handler, if the permissions do not include the permissions required by the ZoneIdentityPermissionAttribute. The ZoneIdentityPermissions are hierarchical, unless customized.
14. A process as in claim 13 in which said action includes terminating said process prior to said executing step.	CLR/CCLR may terminate the process or transfer control to an exception handler that may itself terminate the process.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,917,912

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
8.	Products infringing include Windows Installer SDK, and products that include the Windows Installer technology.
A process comprising the following steps:	Scenario 1: use of Windows Installer packages (i.e. .MSI files) to create Windows Installer-enabled applications, such as Office 2000 and used of the WI service to install them. Scenario 2: software distribution technologies that use the Windows Installer OS service for installation, such as Internet Component Download and products like Office Web Components. Either scenario can be used by SMS, IntelliMirror and third party tools like InstallShield and WISE. NT or later operating systems (because they use the subsystem identifier) using cabinet files, .CAB, (because they have a manifest and INF and/or OSD files), and have been signed with a digital signature and will be authenticated by Authenticode or WinVerifyTrust API and contain at least one PE (portable executables)
(a) accessing a first record containing information directly or indirectly identifying one or more elements of a first component assembly,	Scenario 1: First record is the .MSI file that contains information on what goes in the assembly and how to install the assembly. Scenario 2: A. First record is the cabinet manifest (indirect instructions) B. Or, First record can be INF and/or OSD files (direct instructions)
(1) at least one of said elements including at least some executable programming,	Both scenarios: The PE (portable executable) in the cabinet file is the executable programming.
(2) at least one of said elements constituting a load module,	Both scenarios: PE is a load module:
(i) said load module including executable programming and a	Both scenarios: The PE has several headers.

header;	
(ii) said header including an execution space identifier identifying at least one aspect of an execution space required for use and/or execution of the load module associated with said header;	Both scenarios: SUBSYTEM is a field in the PE Optional Header that is an execution space
(iii) said execution space identifier provides the capability for distinguishing between execution spaces providing a higher level of security and execution spaces providing a lower level of security;	Both scenarios: SUBSYSTEM distinguishes between programs that can run in kernel mode and those that can run in user mode. This is a key security concept of process separation that was introduced with Windows NT. The Subsystem field in the PE header is used by the system to indicate whether the executable will run within Ring 3 (user mode) or use Ring 0 (native or kernel mode). Anything running in Ring 3 is limited to its own processing space. Executables running in Ring 0 can reach out to other spaces and have security measure built around them.
(b) using said information to identify and locate said one or more elements;	Scenario 1: the MSI file identifies and locates the elements Scenario 2: .CAB manifest is used to identify Physical location OSD and/or INF is used to identify Logical location
(c) accessing said located one or more elements;	Scenario 1: Using the MSI file Scenario 2: Using INF and/or OSD in cabinet file
(d) securely assembling said one or more elements to form at least a portion of said first component assembly;	Both scenarios: Using the Window Installer OS service with various properties and flags on the settings for higher protection. Windows Installer has numerous flags that the developer can set to indicate how the assembly will be installed, in what privilege level, with how much user interface, and how much ability the user has to watch or change what is occurring. These controls have been strengthened with each release of Windows Installer. Windows Installer 1.1 and later has the ability to limit the users capabilities during the installation. In a Windows 2000

environment and later, using the Group Policy-based Change and Configuration Management, the administrator has the most control

Fields that can be set by the developer or administrator to control what users can do include the following:

Transformssecure can be set to a value of 1 to inform the installer that transforms are to be cached locally on the user's computer in a location the user does not have write access. (Transforms create custom installations from a basic generic installation, for example to make the Finance versions different from the Marketing version or English versions different from Japanese versions.)

AllowLockdownBrowse and *DisableBrowse* can prevent users from browsing to the sources.

SourceList can be used to specify the only allowable source to be used for the installation of a given component.

Environment can be used to specify whether the installation can be done while the user is logged on or only when no user is logged on.

Security Summary Property conveys whether a package can be opened as read-only or with no restriction.

Privileged Property is used by developers of installer packages to make the installation conditional upon system policy, the user being an administrator, or assignment by an administrator.

Restricted Public Properties can be set as variables for an installation. "For managed installations, the package author may need to limit which public properties are passed to the server side and can be changed by a user that is not a system administrator. Some are commonly necessary to maintain a secure environment when the installation requires the installer use elevated privileges. " *SecureCustomProperties* can be created by the author of an installation package to add controls beyond the default list.

MsiSetInternalUI specifies the level of user interface from none to full.

A *Sequence Table* can be used to specify the required order of execution for the installation process. There are three modes, one of which is the *Administrative Installation* that is used by the network administrator to assign and install applications.

InstallServicesAction registers a service for the system and it can only be used if the user is

	an administrator or has elevated privileges with permission to install services or that the application is part of a managed installation. *DisableMedia* system policy disables media sources and disables browsing to media sources. It can be used with *DisableBrowse* to secure installations version 1.1 that doesn't have some of the other capabilities. *AlwaysInstallElevated* can be set per user or per machine and is used to install managed applications with elevated privileges. AllowLockdownBrowse, AllowLockdownMedia and AllowLockdownPatch set these capabilities so they can only be performed by an administrator during an elevated installation. [See article "HowTo: Configure Windows Installer for Maximum Security (Q247528). Windows XP Professional and .NET have the additional capability to set *Software Restriction Policies* and have these used by Windows Installer. In addition, most of the software distribution technologies that use Windows Installer also add a layer of their own controls. For example, SMS 2.0 enables the administrators to control the installation is optional or required and whether the user can affect the installation contents/features at all.
(e) executing at least some of said executable programming; and	Both scenarios: Part of executable is called during installation in order to do self-registration or perform custom actions. The overall executable is used at runtime.
(f) checking said record for validity prior to performing said executing step.	Scenario 1: Sign the overall package and the cabinet files. Scenario 2: The cabinet file is signed. For IE with the default security level or higher, the digital signature is verified by Authenticode or a similar utility before the component is allowed to be assembled.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,917,912

35.	Products infringing include all products that host the Microsoft .NET Common Language Runtime or Compact Common Language Runtime.
A process comprising the following steps:	
(a) at a first processing environment receiving a first record from a second processing environment remote from said first processing environment;	Computer running the Microsoft CLR/CCLR receives, for example, a shared assembly header or a complete shared assembly from another computer, for example a server.
(1) said first record being received in a secure container;	The shared assembly is cryptographically hashed and signed.
(2) said first record containing identification information directly or indirectly identifying one or more elements of a first component assembly;	The first record is either an assembly manifest, or a whole assembly; the elements are other assemblies that are referenced as external in the first record; the first component assembly is a .NET application domain.
(i) at least one of said elements including at least some executable programming;	Assembly contains executable programming.
(ii) said component assembly allowing access to or use of specified information;	The specified information can include any kind of data file, stream, log, environment variables, etc.
(3) said secure container also including a first of said elements;	The shared assembly includes at least some executable programming.
(b) accessing said first record	CLR/CCLR accesses the assembly or assembly header.
(c) using said identification information to identify and locate said one or more elements;	Manifest and type metadata information section is used to identify and locate files, code elements, resource elements, individual classes and methods.
(1) said locating step including locating a second of said elements at a third processing environment located remotely from said first processing environment and said second processing environment;	Met by a multifile assembly, with files distributed across a network, or by the second element constituting another referenced assembly located elsewhere; the CLR/CCLR uses probing to locate and access the file.
(d) accessing said located one or more elements;	Step carried out by the CLR/CCLR loader.
(1) said element accessing step including retrieving said second element from said third processing environment;	Step carried out by the CLR/CCLR loader.
(e) securely assembling said one or more elements to form at least a portion of said first component assembly specified by said first record; and	CLR/CCLR carries out this step, including checking the integrity of the load module, checking the load module's permissions, placing the load module contents into an application domain, isolating it from malicious or badly behaved code, and from code that

	does not have the permission to call it.
(f) executing at least some of said executable programming,	Step carried out by the CLR/CCLR.
(1) said executing step taking place at said first processing environment.	CLR/CCLR is operating in the first processing environment specified above.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,920,861

34.	Product Infringing: Microsoft Operating Systems that support device driver signature technology
A descriptive data structure embodied on a computer-readable medium or other logic device including the following elements:	
a representation of the format of data contained in a first rights management data structure	The driver package's INF is a data structure. The INF contains multiple types of sections, structured as hierarchy /"branches," that the Windows operating system or its Plug and Play and/or Set-up installation services "branch" through based on the operating system information and device for which a driver is to be installed. The installation services use the "branching" structure (format) to determine what files should be installed. The INF, further provides disk location information and file directory path information for the files identified as necessary as a result of the "branching" process. The driver package is a "rights management" data structure based on the fact that it is governed and based on the fact that it processes governed information. Rights Management as Governed Item A driver manufacturer can include rules governing the driver's installation and/or use in the driver's INF file. For example: Security entries specify an access control list for the driver. Driver developers can specify rules that determine behavior of the driver package based on the user's operating system version, including product type and suite and the device for which the driver is to be installed Rules specifying logging Local administrators can establish policy as to what action or notification should occur in the event that a driver being installed is not signed.

	The operating system installation services have a ranking criteria it follows when multiple drivers are available for a newly detected device. The criterion is used to determine the driver best suited for ensuring compatibility with the operating system and ensuring functionality of the device. Drivers have been certified to be compatible with specified operating system versions for their respective device classes. The catalog file protects the integrity of the driver. Microsoft distributes the Driver Protection List to prevent known bad deriver from being installed. Processing Rights Managed Items Certain drivers (SAP) have been explicitly certified to protect DRM content. MSDN – DRM Overview A DRM-compliant driver must prevent unauthorized copying while digital content is being played. In addition, the driver must disable all digital outputs that can transmit the content over a standard interface (such as S/PDIF) through which the decrypted content can be captured.
said representation including:	
element information contained within said first rights management data structure; and	The elements of a driver package include: A driver that is typically a dynamic-link library with the *.sys* filename extension. An INF file containing information that the system Setup components use to install support for the device. A driver catalog file containing the digital signature. One or more optional co-installers which are a Win32® DLL that assists in device installation NT-based operating systems. Other files, such as a device installation application, a device icon, and so forth. XP DDK – INF Version Section The LayoutFile entry specifies one or more additional system-supplied INF files that contain layout information on the source media required for installing the software

	described in this INF. All system-supplied INF files specify this entry. The CatalogFile entry specifies a catalog (*.cat*) file to be included on the distribution media of a device/driver.
organization information regarding the organization of said elements within said first rights management data structure; and	Within an INF is a hierarchy with the top being a list of manufacturers, and sub-lists of models and at the bottom a list of install information by model. For Windows XP and later versions of NT-based operating systems, entries in the **Manufacturer** section can be decorated to specify operating system versions. The specified versions indicate OS versions with which the specified INF *Models* sections will be used. If no versions are specified, Setup uses the specified *Models* section for all versions of all operating systems. INF's SourceDisksNames and SourceDisksFiles sections specify organization information. **XP DDK -- Source Media for INFs** The methods you should use to specify source media for device files depend on whether your INFs ship separately from the operating system or are included with the operating system. INFs for drivers that are delivered separately from the operating system specify where the files are located using SourceDisksNames and SourceDisksFiles sections. If the files to support the device are included with the operating system, the INF must specify a **LayoutFile** entry in the Version section of the file. Such an entry specifies where the files reside on the operating system media. An INF that specifies a **LayoutFile** entry must not include **SourceDisksNames** and **SourceDisksFiles** sections. **XP DDK – INF SourceDisksNames Section** A **SourceDisksNames** section identifies the distribution disks or CD-ROM discs that contain the source files to be transferred to the target machine during installation. Relevant values of an entry in the INF include: *diskid* -- Specifies a source disk. *disk-description* – Describes the contents

	and/or purpose of the disk identified by *diskid*. *tag-or-cab-file* -- This optional value specifies the name of a tag file or cabinet file supplied on the distribution disk, either in the installation root or in the subdirectory specified by *path*, if any. *path* -- This optional value specifies the path to the directory on the distribution disk containing source files. The *path* is relative to the installation root and is expressed as *\dirname1\dirname2...* and so forth. *flags* -- For Windows XP and later, setting this to **0x10** forces Setup to use *cab-or-tag-file* as a cabinet file name, and to use *tag-file* as a tag file name. Otherwise, *flags* is for internal use only. *tag-file* -- For Windows XP and later, if *flags* is set to **0x10**, this optional value specifies the name of a tag file supplied on the distribution medium, either in the installation root or in the subdirectory specified by *path*. The value should specify the file name and extension without path information. XP DDK -- INF SourceDisksFiles Section A **SourceDisksFiles** section names the source files used during installation, identifies the source disks (or CD-ROM discs) that contain those files, and provides the path to the subdirectories, if any, on the distribution disks containing individual files. Relevant values in an entry in the INF would include: *filename* -- Specifies the name of the file on the source disk. *diskid* -- Specifies the integer identifying the source disk that contains the file. This value and the initial *path* to the *subdir*(ectory), if any, containing the named file must be defined in a **SourceDisksNames** section of the same INF. *subdir* -- This optional value specifies the subdirectory (relative to the **SourceDisksNames** *path* specification, if any) on the source disk where the named file resides.
information relating to metadata, said metadata including:	
metadata rules used at least in part to govern at least one aspect of use and/or display of content stored within a rights management data structure,	The driver manufacture can specify rules in the INF that govern the installation and/or use of the driver. For example, security entries specify an access control list for the

	driver. Driver developers can specify rules in an INF file that determines behavior of the driver package based on the user's operating system version, including product type and suite. Also, rules related to logging can be specified as mentioned in next claim element. **For Example – Access Control List Rules** **XP DDK – Tightening File-Open Security in a Device INF File** For Microsoft Windows 2000 and later, Microsoft tightened file-open security in the class installer INFs for certain device classes, including CDROM, DiskDrive, FDC, FloppyDisk, HDC, and SCSIAdapter. If you are unsure whether the class installer for your device has tightened security on file opens, you should tighten security by using the device's INF file to assign a value to the **DeviceCharacteristics** value name in the registry. Do this within an *add-registry-section*, which is specified using the INF AddReg directive. XP-DDK -- INF AddReg Directive An INF can also contain one or more optional *add-registry-section*.**security** sections, each specifying a security descriptor that will be applied to all registry values described within a named *add-registry-section*. A **Security** entry specifies a security descriptor for the device. The *security-descriptor-string* is a string with tokens to indicate the DACL (**D:**) security component. A class-installer INF can specify a security descriptor for a device class. A device INF can specify a security descriptor for an individual device, overriding the security for the class. If the class and/or device INF specifies a *security-descriptor-string*, the PnP Manager propagates the descriptor to all the device objects for a device, including the FDO, filter DOs, and the PDO. For Example – Operating System Versioning Operating-System Versioning for Drivers

	under Windows XP Setup selects the [*Models*] section to use based on the following rules: If the INF contains [*Models*] sections for several major or minor operating system version numbers, Setup uses the section with the highest version numbers that are not higher than the operating system version on which the installation is taking place. If the INF [*Models*] sections that match the operating system version also include product type decorations, product suite decorations, or both, then Setup selects the section that most closely matches the running operating system.
said metadata rules including at least one rule specifying that information relating to at least one use or display of said content be recorded and/or reported.	The AddService directive can set up event-logging services for drivers. **INF AddService Directive** An **AddService** directive is used to control how (and when) the services of particular Windows 2000 or later device's drivers are loaded, any dependencies on other underlying legacy drivers or services, and so forth. Optionally, this directive sets up event-logging services by the devices/drivers as well. Relevant sections of the directive's entry include: *event-log-install-section* -Optionally references an INF-writer-defined section in which event-logging services for this device (or devices) are set up. *EventLogType* -- Optionally specifies one of **System**, **Security**, or **Application**. If omitted, this defaults to **System**, which is almost always the appropriate value for the installation of device drivers. For example, an INF would specify **Security** only if the to-be-installed driver provides its own security support. *EventName* -- Optionally specifies a name to use for the event log. If omitted, this defaults to the given *ServiceName*.

35. A descriptive data structure as in claim 34, in which:	
said first rights management data structure comprises a first secure container.	The driver package is secured through a catalog file that is signed by Microsoft's Windows Hardware Quality Lab and

	contains the hash of each file of the driver's package. The INF identifies the catalog file used to sign the driver package.
36. A descriptive data structure as in claim 35, in which:	
said first secure container comprises:	The first secure container is the driver package secured by a catalog file.
said content; and	The content is the driver and related files within the signed driver package.
rules at least in part governing at least one use of said content.	The rules are within the INF, which is part of the signed driver package.
37. A descriptive data structure as in claim 36, wherein the descriptive data structure is stored in said first secure container.	The INF is stored within the signed driver package.
44. A descriptive data structure as in claim 34, further including:	
a representation of the format of data contained in a second rights management data structure,	The manufacture and models sections in the INF Version section are provided for the possibility of a single INF representing the format for multiple drivers. Operating system version "decorating" relating the architecture, major and minor operating systems versions, product and suit information all relate to the target environment and is used to identify the files necessary for the target environment. An INF file, such as in the case of operating system targeting, can be used for more than one driver package since it can contain more than one catalog file. Further an INF can address the drives necessary for a multi-functional device.
said second rights management data structure differing in at least one respect from said first rights management data structure.	The files of the second data structure would vary from the files on the first data structure.
45. A descriptive data structure as in claim 44, in which:	
said information regarding elements contained within said first rights management data structure includes information relating to the location of at least one such element.	INF specify where the driver files are located using the SourceDiskNames and SourceDiskFiles sections.
46. A descriptive data structure as in claim 44, further including:	
a first target data block including information relating to a first target	Operating system version "decorating" relating the architecture, major and minor

environment in which the descriptive data structure may be used.	operating systems versions, product and suit information all relate to the first target environment.

47. A descriptive data structure as in claim 46, further including:	
a second target data block including information relating to a second target environment in which the descriptive data structure may be used,	Operating system version decorating will cover multiple operating systems.
said second target environment differing in at least one respect from said first target environment.	This is the reason for version decorating.

48. A descriptive data structure as in claim 46, further including:	
a source message field containing information at least in part identifying the source for the descriptive data structure.	The provider entry in the version section of the INF identifies the provider of the INF file. Also, the INF contains a manufacture section.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,920,861

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
58.	Product Infringing: Microsoft Reader SDK and Microsoft Digital Asset Server.
A method of creating a first secure container, said method including the following steps;	Method is carried out by Microsoft's Digital Asset Server and Microsoft's .Litgen tools
(a) accessing a descriptive data structure, said descriptive data structure including or addressing	.opf file describing the file structure of a protected e-book including metadata, manifest, and "spine" information
(1) organization information at least in part describing a required or desired organization of a content section of said first secure container, and	Organization information regarding organization of the ebook and the inscription as specified in the manifest and spine information in the .opf file
(2) metadata information at least in part specifying at least one step required or desired in creation of said first secure container;	Metadata constitutes rules specifying the degree of security to use and/or XrML rules
(b) using said descriptive data structure to organize said first secure container contents	e-book packaging carried out by Microsoft Litgen tool
(c) using said metadata information to at least in part determine specific information required to be included in said first secure container contents; and	Step performed by Digital Asset Server; example of specific information is owner/purchaser information required in the inscription process
(d) generating or identifying at least one rule designed to control at least one aspect of access to or use of at least a portion of said first secure container contents.	Analyzing the metadata and finally packaging the e-book using a particular security level specified through the metadata
71. A method as in claim 58, in which:	
(a) said specific information required to be included includes information at least in part identifying at least one owner or creator of at least a portion of said first secure container contents.	Owner purchaser information required in the inscription process; XrML rule requiring display of copyright notice

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,920,861

58.	Product Infringing: All products that host the Microsoft Common Language Runtime or Compact Common Language Runtime.
A method of creating a first secure container, said method including the following steps;	Method is practiced by a user using the Common Language Runtime (CLR) or Compact Common Language Runtime (CCLR) to create a dynamic shared assembly or .NET Framework SDK to create a shared assembly
(a) accessing a descriptive data structure, said descriptive data structure including or addressing	.NET framework Assembly class and/or AssemblyBuilder class and/or AssemblyInfo file
(1) organization information at least in part describing a required or desired organization of a content section of said first secure container, and	This information is specified in the classes named above and in the AssemblyInfo file.
(2) metadata information at least in part specifying at least one step required or desired in creation of said first secure container;	This information is addressed in the classes and the AssemblyInfo file, e.g., for a shared assembly metadata will be specified that the assembly is to be signed using specified key
(b) using said descriptive data structure to organize said first secure container contents;	This step is carried out by applications and tools using the classes and assembly info file, including CLR (or CCLR) and .NET Framework SDK
(c) using said metadata information to at least in part determine specific information required to be included in said first secure container contents; and	This step is carried out by applications and tools using the assembly info file and classes that specify the metadata required in the target assembly
(d) generating or identifying at least one rule designed to control at least one aspect of access to or use of at least a portion of said first secure container contents.	User may specify rules, as specified in the .NET Framework SDK, to be placed in the assembly manifest including such rules requiring that all code be managed (CLR or CCLR compliant), "Code Access Security" permissions be supplied for use of code supplied in the assembly, etc
64. A method as in claim 58, in which:	
(a) said creation of said first secure container occurs at a first data processing arrangement located at a first site;	Can be a server, PC or workstation running CLR (or CCLR) to create a dynamic shared assembly or .NET Framework SDK to create a shared assembly)
(b) said first data processing arrangement including a communications port; and	Included in virtually any computer
(c) said method further includes:	
(1) prior to said step of accessing said descriptive data structure, said	Download of the assemblyinfo file and/or a file containing a class calling the

first data processing arrangement receiving said descriptive data structure from a second data processing arrangement located at a second site,	DefineDynamicAssembly methods or download of SDK containing assemblybuilder class from a second site
(d) said receipt occurring through said first data processing arrangement communications port.	Communications port is normally used for downloading
67. A method as in claim 64, further comprising:	
at said first processing site, receiving said metadata through said communications port.	Download of the AssemblyInfo file and/or a file containing a class calling the DefineDynamicAssembly methods or download of SDK containing assemblybuilder class from a second site
68. A method as in claim 67, in which,	
(a) said metadata is received separately from said descriptive data structure.	Method practiced when metadata names are addressed by the assembly class and a template for the AssemblyInfo file, and values corresponding to those names are received through a user interface such as provided by Microsoft Visual Studio or are provided from a separate file
71. A method as in claim 58, in which:	
(a) said specific information required to be included includes information at least in part identifying at least one owner or creator of at least a portion of said first secure container contents.	The Assembly class definition includes attributes for company name and trademark information, and these may be required attributes specified in the AssemblyInfo file
72. A method as in claim 58, in which:	
(a) said specific information required to be included includes a copyright notice.	The Assembly class definition includes an attribute for copyright field that may be required by the AssemblyInfo file

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,920,861

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
58.	Product Infringing: Microsoft .NET Framework, Visual Studio .NET, and tools that include the Assembly Generator tool AL.exe.
A method of creating a first secure container, said method including the following steps;	The **Assembly Generation tool** generates a portable execution file with an assembly manifest from one or more files that are either Microsoft intermediate language (MSIL) modules or resource files. When using the tool's signing option, the assembly becomes a *secure container*.
(a) accessing a descriptive data structure, said descriptive data structure including or addressing	The *descriptive data structure* is the text file used as input by the Assembly Generation tool.
(1) organization information at least in part describing a required or desired organization of a content section of said first secure container, and	The DDS specifies the *link* and or *embed* directives to indicate which source files should be included in the assembly, how the included resource will be tagged, and if the resource will be private. Private resources are not visible to other assemblies. These tags are used to organize the assembly into *named* sections. Private attributes are used to organize the assembly into both public and *private* sections. (Public sections are the default.)
(2) metadata information at least in part specifying at least one step required or desired in creation of said first secure container;	The text file can contain "options" relating to how the assembly should be built and additional information that should be included. *Main* – Specifies the method to use as an entry point when converting a module to an executable file. *Algid* – Specifies an algorithm to hash all files. *Comp* – Specifies string for the Company field. *Conf* – Specifies string for Configuration field *Copy* – Specifies string for Copyright field. *Culture* – Specifies the culture string to associate with the assembly. *Delay* – Variation of this option specifies whether the assembly will be

	fully or partially signed and whether the public key is placed in the assembly. *Description* – Specifies the description field. *Evidence* – Embeds file in the assembly with the resource name Security.Evidence. *Fileversion* – Specifies the file version of the assembly. *Flags* – Specifies flags for such things as the assembly is side-by-side compatible, assembly cannot execute with other versions if either they are executing in the same application domain, process or computer. *Keyf* – Specifies a file that contains a key or key pair to sign an assembly. *Keyn* – Specifies the container that holds a key pair. *Product* – Specifies string for Product field. *Productv* – Specifies string for Product Version. *Template* – Specifies the assembly fro which to inherit all assembly metadata. *Title* – Specifies string for Title field. *Trade* – Specifics string for Trademark field. *V* – Specifies version information.
(b) using said descriptive data structure to organize said first secure container contents	The following directives are used to specify which files are to be compiled into the assembly, how they will be tagged, and whether or not they will be visible to other assemblies, AKA private: *Embed[*name, *private]* – copies the content of the file into the assembly and applies an optional name tag, and optional private attribute. *Link[*name, *private]* – file becomes part of the assembly via a link and applies an optional name tag, and optional private attribute.
(c) using said metadata information to at least in part determine specific information required to be included in said first secure container contents; and	The following are some of the "options" address what information should be included in the secure container: *Main* – Specifies the method to use as an entry point when converting a module to an executable file. *Comp* – Specifies string for the Company field. *Conf* – Specifies string for Configuration field *Copy* – Specifies string for Copyright

	field. *Culture* – Specifies the culture string to associate with the assembly. *Description* – Specifies the description field. *Evidence* – Embeds file in the assembly with the resource name Security.Evidence. *Fileversion* – Specifies the file version of the assembly. *Flags* – Specifies flags for such things as the assembly is side-by-side compatible, assembly cannot execute with other versions if either they are executing in the same application domain, process or computer. *Keyf* – Specifies a file that contains a key or key pair to sign an assembly. *Keyn* – Specifies the container that holds a key pair. *Product* – Specifies string for Product field. *Productv* – Specifies string for Product Version. *Template* – Specifies the assembly fro which to inherit all assembly metadata. *Title* – Specifies string for Title field. *Trade* – Specifics string for Trademark field. *V* – Specifies version information.
(d) generating or identifying at least one rule designed to control at least one aspect of access to or use of at least a portion of said first secure container contents.	User may specify rules, as specified in the .NET Framework SDK, to be placed in the assembly manifest including such rules requiring that all code be managed (CLR compliant), "Code Access Security" permissions be supplied for use of code supplied in the assembly, etc.
71. A method as in claim 58, in which:	
(a) said specific information required to be included includes information at least in part identifying at least one owner or creator of at least a portion of said first secure container contents.	The following "options" specifies owner and creator information: *Comp* – Specifies string for the Company field. *Copy* – Specifies string for Copyright field. *Trade* – Specifics string for Trademark field.
72. A method as in claim 58, in which:	
(a) said specific information required to be included includes a copyright notice.	The copy "option" specifies the string for the for the Copyright field.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,982,891

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
1.	Products infringing: All products that include the Common Language Runtime or Compact Common Language Runtime or Common Language Infrastructure.
A method for using at least one resource processed in a secure operating environment at a first appliance, said method comprising:	Resource may constitute a Microsoft Windows process or hardware element; secure operating environment is Microsoft Common Language Runtime ("CLR") environment, Common Language Infrastructure ("CLI") or Compact CLR ("CCLR"); first appliance is computer running CLR, CLI or Compact CLR. Two infringing scenarios are set forth herein: (1) For CLR, an administrator, using the .NET framework caspol.exe tool remotely configures security policy in a .NET configuration file for a machine, enterprise, user, or application and that security policy interacts with rules or evidence declared in a shared assembly provided by another entity ("1st scenario"); and (2) for CLR, CLI and CCLR two assemblies are delivered to an appliance; the first assembly has a rule that demands permissions from a caller in the second assembly, and the second assembly includes a control that asserts such permissions or provides evidence that convinces the runtime that it has such permissions. ("2nd scenario"). In each scenario Microsoft .NET "Code Access Security" framework or "Role Based Security" framework is used.
(a) securely receiving a first entity's control at said first appliance, said first entity being located remotely from said operating environment and said first appliance;	1st scenario: first entity is the administrator, and the policy that constitutes this entity's control is securely received at the first appliance through a session established between the administrator's computer and the first appliance, requiring security credentials such as the administrator's login and password or other secure session means. 2nd scenario: first entity is creator or distributor of the first assembly, assembly manifest includes a control demanding or refusing or otherwise asserting a security action on permissions from a caller; first assembly is integrity-checked.
(b) securely receiving a second entity's control at said first appliance, said second entity being located remotely from said operating environment and said first appliance, said second entity being different from said first	Second entity's control is contained in shared assembly manifest (and therefore integrity protected) that provides evidence for obtaining permissions, or asserts permissions; assembly creator/distributor is located remotely and is

entity; and	not the administrator (1st scenario) or creator/distributor of the first container (2nd scenario);
(c) securely processing a data item at said first appliance, using at least one resource, including securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item.	Secure processing is carried out by CLR, CLI or CCLR, Data item constitutes an executable code element, an interface controlled by such an executable, a data collection or stream (such as media file or stream or text file) or an environment variable. CLR, CLI or CCLR securely processes the rules, which will in both scenarios govern access to methods and data from the first assembly. The resource named in the claim is, e.g., a Windows process that is established by the runtime or hardware element on the computer.
51. A method as in claim 1 wherein at least said secure processing step is performed at an end user electronic appliance.	Consumer computer or appliance running Microsoft CLR, CLI or CCLR).
58. A method as in claim 1 wherein the step of securely receiving a first entity's control comprises securely receiving said first entity's control from a remote location over a telecommunications link, and the step of securely receiving said second entity's control comprises securely receiving said second entity's control from the same or different remote location over the same or different telecommunications link.	1st scenario 1: link is LAN or WAN; 2nd scenario: link is any telecommunications link, including the internet.
65. A method as in claim 1 wherein the processing step includes processing said first and second controls within the same secure processing environment.	Secure processing environment is CLR, CLI or CCLR running on user's computer or appliance.
71. A method as in claim 1 further including the step of securely combining said first entity's control and said second entity's control to provide a combined control arrangement.	In scenario 2, arrangement consists of the stack frame, and the corresponding array of permission grants for assemblies on the stack, and the permission demanded by the first assembly. Secure combining performed by the CLR, CLI or CCLR.
76. A method as in claim 1 wherein said two securely receiving steps are independently performed at different times.	Steps are performed at different times in both scenarios.
84. A method as in claim 1 wherein at least one of the first entity's control and the second entity's control comprises at least one executable component and at least one data component.	In both scenarios the second entity supplies an assembly with a demand procedure executed by the CLR, CLI or CCLR. The data component is a specific attribute value referenced by the assembly.
89. A method as in claim 1 wherein said first appliance includes a protected processing environment, and wherein:	Microsoft Common Language Runtime (CLR), Common Language Infrastructure (CLI), or Compact Common Language Runtime (CCLR) environment.
(a) said method further comprises a step of receiving, at said first appliance, said data item	Typically occurs in both scenarios.

separately and at a different time from said receiving said first entity's control ; and	
(b) said securely processing step is performed at least in part in said protected processing environment	Protected processing environment is the CLR, CLI or CCLR.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,982,891

22.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A method of securely controlling use by a third party of at least one protected operation with respect to a data item comprising:	A user (third party) accesses an IRM-protected data item governed by IRM controls under two or more RMS servers. For example, the data item may be a IRM-protected document. The IRM controls may be associated with the data item directly or via a IRM-protected container holding the IRM-protected data item, such as an IRM-protected email with the IRM-protected document attached.
(a) supplying at least a first control from a first party to said third party;	The user acquires a first use license from a first RMS server (first party) enabling access to, the IRM-protected data item under the IRM rules associated with the first RMS server. For example: (1) the first use license from the first RMS server permits the user to access a IRM-protected document contained within or attached to an IRM-protected email; or (2) the first use license from the first RMS server applies a first set of IRM rules to an IRM-protected document.
(b) supplying, to said third party, at least a second control from a second party different from said first party;	The user acquires a second use license from a second RMS server (second party) enabling access to the IRM-protected data item under the IRM rules associated with the second RMS server. For example: (1) in addition to the user being given access to an IRM-protected email based on a first use license, a second RMS server provides a second use license enabling access to the IRM-protected document attached thereto; or (2) the second use license from the second RMS server applies a second set of IRM rules to the IRM-protected document.
(c) securely combining at said third party's location, said first and second controls to form a control arrangement;	The first and second use licenses are combined to form a control arrangement that governs access to the IRM-protected data item.
(d) securely requiring use of said control arrangement in order to perform at least one protected operation using said data item; and	The combined first and second use licenses govern access to the IRM-protected data item.
(e) securely performing said at least one protected operation on behalf of said third party with respect to said data item by at least in part employing said control arrangement	The user performs a protected operation (e.g., read, print, edit) on the IRM-protected data item. The combined first and second use licenses are employed to permit the protected operation.

23. A method as in claim 22 wherein said data item is protected.	The data item is encrypted and protected by IRM.
39. A method as in claim 22 further including securely and persistently associating at least one of: (a) said first control, (b) said second control, and (c) said control arrangement, with said data item.	The first and/or second use license are securely and persistently associated with the IRM-protected data item.
53. A method as in claim 22 wherein at least two of the recited steps are performed at an end user electronic appliance.	Steps performed at a user's computer or appliance.
60. A method as in claim 22 wherein step (a) comprises supplying said first control from at least one remote location over a telecommunications link, and step (b) comprises supplying said second control from the same or different remote location over the same or different telecommunications link	The first and second use licenses are received over a telecommunications link such as a networking or modem/serial interface.
67. A method as in claim 22 wherein at least step (c) is performed within the same secure processing environment at said third party's location.	Steps are performed at user's computer or appliance.
91. A method as in claim 22 wherein:	
(a) said method further comprises supplying said data item to said third party separately and at a different time from supplying of said first control to said third party; and	The first use license (first control) is received at the time that the user accesses the data item, which occurs separately and at a different time from receipt of the IRM-protected data item itself.
(b) said securely performing step comprises performing said protected operation at least in part in a protected processing environment.	The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-debugging, and tamper resistance.

26.	Products infringing: Visual Studio.NET, .NET Framework SDK, and all products that include the Common Language Runtime or Compact Common Language Runtime or Common Language Infrastructure.
A secure method for combining data items into a composite data item comprising:	
(a) securely providing, from a first location to a second location, a first data item having at least a first control associated therewith;	A first signed and licensed .NET component, .NET assembly, managed control and/or Web control (component) is the first data item. The first .NET component developer (first location) provides the application assembly developer (second location) the first component. The first control is the set of declarative statements comprising the LicenseProviderAttribute (alternately referred to as license controls).
(b) securely providing, from a third location to said second location, a second data item having at least a second control associated therewith;	A second signed and licensed component is the second data item. The second component developer (third location) provides the application assembly developer (second location) the second component. The second control is the set of declarative statements comprising the LicenseProviderAttribute.
(c) forming, at said second location, a composite of said first and second data items;	The application assembly developer will include at least the two components into its assembly.
(d) securely combining. at said second location, said first and second controls to form a control arrangement; and	At the second location, the application assembly developer uses the .NET runtime that includes the LicenseManager. Whenever a component is instantiated (here, an instance of the first licensed component), the license manager accesses the proper validation mechanism for the component. The license controls (first control) for the runtime license (derived from the design-time license) are bound into the header of the .NET application assembly, along with the second control for the second component. Visual Studio.NET securely handles the creation of runtime license controls. Runtime licenses are embedded into (and bound to) the executing application assembly. The license control attribute

	included in the first component is customized in the second location to express and require the runtime license. In a more advanced scenario, the License Complier tool can be used to create a ".licenses file" containing licenses for multiple components, including runtime licenses for components and classes created by the license provider. This .licenses file is embedded into the assembly. The third control set comprises the runtime license controls for the first and second components (that had been bound to the assembly), the declarative controls provided by the application assembly developer, and any runtime licenses for other components included by the developer in application assembly. The controls are typically integrated into the header of the .NET application assembly calling the first licensed component.
(e) performing at least one operation on said composite of said first and second data items based at least in part on said control arrangement.	The proper execution of the application will require that the assembly have run time licenses for the two components.
27. A method as in claim 26 wherein said combining step includes preserving each of said first and second controls in said composite set.	The set of declarative statements comprising the LicenseProviderAttribute of both the first and second components are included in the application assembly.
28. A method as in claim 26 wherein said performing step comprises governing the operation on said composite of said first and second data items in accordance with said first control and said second control.	The application will require the first and second controls to operate properly when it calls the first and second data items, respectively.
29. A method as in claim 26 wherein said providing step includes ensuring the integrity of said association between said first controls and said first data item is maintained during at least one of transmission, storage and processing of said first data item.	Signing the component that has embedded within it the license control ensures the integrity of the association of the control and data item.
31. A method as in claim 26 wherein said providing step comprises codelivering said first data item and said first control.	The component includes the license control and therefore they are codelivered.
40. A method as in claim 26 further including the step of securely ensuring that at least one of (a) said first control, (b) said second control, and (c) said control arrangement, is persistently associated with	Each component includes the license control. Signing the component that has embedded within it the license control ensures the persistence of the association of the control and data item.

at least one of said first and second data items.	
54. A method as in claim 26 wherein at least one of steps (c), (d) and (e) is performed at an end user electronic appliance.	At least step (e) is typically performed at an end-user electronic appliance.
61. A method as in claim 26 wherein step (a) comprises providing said first data item from at least one remote location over a telecommunications link, and step (b) comprises providing said second data item from the same or different remote location over the same or different telecommunications link.	Microsoft maintains Web sites where a developer can get components over the Web. These sites include references whereby a developer may obtain components through their Web connection. One such site is Internet Explorer Web Control Gallery at ie.components.microsoft.com/webcontrols
68. A method as in claim 26 wherein step (d) is performed within the same secure processing environment at said second location.	Typically, step (d) will be performed within the same secure processing environment.
79. A method as in claim 26 wherein steps (a) and (b) are performed at different times.	The application assembly developer will typically acquire components at different times.
86. A method as in claim 26 wherein at least one of the first and second controls comprises at least one executable component and at least one data component.	The component must include an executable and can include a data items as a EULA, readme file or help file.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,982,891

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
35	Infringing products include: Windows Media Player, Individualized DRM Clients and the Secure Audio Path (SAP) technology.
A method for using at least one resource processed by a secure operating environment, said method comprising:	
securely receiving a first load module provided by a first entity external to said operating environment	The Individualized DRM Client (first load module) is a signed security upgrade DLL. It is also bound to the hardware ID of the machine on which it runs. It is therefore securely delivered and integrity protected.
securely receiving a second load module provided by a second entity external to said operating environment, said second entity being different from said first entity; and	A SAP certified driver is also signed and carries with it a certificate that indicates its compliance with SAP criteria. If it is delivered to a PC it is secure in the sense that it is integrity protected. This driver would not come from the same entity as the Individualization DLL.
securely processing, using at least one resource, a data item associated with said first and second load modules, including securely applying said first and second load modules to manage use of said data item.	If a WM audio file targeted to the Individualized DRM client carries with it a requirement that SAP be supported to render the WMF contents, the content is processed for playing through a soundcard using the WMP and by applying the DRM client - which decrypts the content and negotiates with the DRM kernel processing of the content through a Secure Audio Path that includes the SAP-certified audio driver.
56. A method as in **claim 35** wherein at least two of the recited steps are performed at an end user electronic appliance.	All steps occur at the user's PC that supports the WMP and DRM client and SAP.
63. A method as in **claim 35** wherein said first load module receiving step comprises securely receiving said first load module from at least one remote location over at least one telecommunications link, and said second load module receiving step comprises securely receiving said second load module from the same or different remote location over the same or different telecommunications link.	The Driver and DRM client are received from distinct locations and may be delivered securely over the Internet. They are delivered securely in that each is integrity protected.
70. A method as in **claim 35** wherein said securely processing step comprises securely executing said first and second	Both load modules are executed on the PC within the WMP/DRM Client/SAP environment.

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
load modules within the same secure processing environment.	
74. A method as in **claim 35** further including securely combining said first and second load modules to provide a combined executable.	Since both the DRM client and the driver are DLLs in the same audio rendering chain, they exist as an execution environment.
81. A method as in **claim 35** wherein said securely receiving steps are performed independently at different times.	The driver and Individualization DLL need not be received at the same time.
94. A method as in **claim 35** wherein said secure operating environment includes a protected processing environment, and wherein: said method further comprises receiving a data item within said secure operating environment; said first load module receiving step is performed separately and at a time different from receiving said data item; and said securely processing step is performed at least in part in said protected processing environment.	The Windows Media Player together with the Individualized DRM Client and Secure Audio Path comprise a protected environment for processing protected media. The protected Windows Media Files are received after the load modules have been received and installed (licenses cannot be acquired until load modules are in place). The processing of the Windows Media File occurs in the protected environment.

Examples of SAP-certified drivers include - as indicated at http://www.microsoft.com/Windows/windowsmedia/WM7/DRM/FAQ.asp#Security7

- All VIA controllers with AC-97 codecs
- All ALI controllers with AC-97 codec
- Intel ICH controllers with AC-97 codecs
- Creative Labs SoundBlaster16/AWE32/AWE64/Vibra
- Yamaha OPL3
- Yamaha DS-1
- Cirrus Logic (Crystal) CS4280
- Cirrus Logic (Crystal) CS4614 / CS4624
- ESS Maestro 2E
- USB Audio
- Cirrus Logic (Crystal) CS4281

- All SiS controllers with AC-97 codecs
- Ensoniq ES1370
- NeoMagic NM6
- Ensoniq ES1371/73 and CT5880
- SoundBlaster Live!
- Aureal 8810
- Aureal 8820
- Aureal 8830
- Conexant Riptide
- ESS Maestro
- ESS ISA parts
- NeoMagic NM5

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,982,891

36.	Product Infringing: Any product using Common Language Runtime (CLR), Common Language Infrastructure (CLI), or Compact Common Language Runtime (CCLR)
A secure operating environment system for managing at least one resource comprising:	Microsoft CLR, CLI or CCLR (operating environment system), managing any of the resources on a typical computer, including memory, files system, communications ports, storage devices, and higher level resources that may use any of these or combinations of them.
(a) a communications arrangement	Communications port and Microsoft Internet Protocol stack that may optionally use Secure Socket Layer protocol or IPSEC packet security protocol, supplied with Microsoft Windows.
(1) that securely receives a first control of a first entity external to said operating environment, and	Rule or evidence contained in the manifest of a shared assembly, distributed by a first entity that can be used by the CLR, CLI or CCLR to determine permissions that may be needed to cause operations on a data item or resource controlled by another entity; shared assembly is tamper-protected and may be received using secure SSL or IPSEC protocol.
(2) securely receives a second control of a second entity external to said operating environment, said second entity being different from said first entity; and	Rule specified in the manifest of a second shared (Tamper protected) assembly, that demands permissions of callers of its methods.
(b) a protected processing environment, operatively connected to said communications arrangement, that:	CLR, CLI or CCLR, connected to (e.g.) communications port
(1) [] securely processes, using at least one resource, a data item logically associated with said first and second controls, and	CLR, CLI or CCLR uses type safety mechanisms, access controls, integrity detection, and separation of domains. Data item may be any data item that is managed by the second assembly, which may be a member of such assembly, and whose state or value may be accessible through an interface to other assemblies, and which is referenced by the first assembly.
(2) [] securely applies said first and second controls to manage said resource for controlling use of said data item.	CLR, CLI or CCLR processes the demand for permissions from the second assembly, collects the evidence or processes the rule from the first assembly, and determines whether the first assembly has the permissions to use the resource to operate on the data item controlled by the second assembly.
57. A system as in claim 36 wherein said protected processing environment is part of an	Computer or electronic appliance running CLR, CLI or CCLR

end user electronic appliance.	
64. A system as in claim 36 wherein said communications arrangement receives said first and second controls from at least one remote location over at least one telecommunications link.	Shared assemblies are designed to be received remotely, e.g., over the internet.
75. A system as in claim 36 wherein said protected processing environment combines said first and second controls to provide a combined control arrangement.	Arrangement consists of the stack frame and and the corresponding array of permission grants for assemblies on the stack, and the permission demanded by the second assembly.
82. A system as in claim 36 wherein said communications arrangement independently receives said first and second controls at different times	Assemblies, including controls, are designed for independent delivery.
88. A system as in claim 36 wherein at least one of the first control and second controls comprises at least one executable component and at least one data component.	The second entity supplies an assembly with a demand procedure (executed by the CLR, CLI or CCLR) that includes reference to a specific attribute value (the data component), and the protected processing environment executes the executable component (demand) in a manner that is at least in part responsive to the data component (execution is in response to the security action supplied in the data item).

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,982,891

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
36.	Infringing Product: My Services
A secure operating environment system for managing at least one resource comprising:	Secure operating environment is the secure server for any .NET My Services service (e.g. My Calendar, My Inbox)
a communications arrangement that securely receives	Secure server receives communications formatted using the SOAP-SEC, the security extension to SOAP that is used by My Service servers to receive controls.
a first control	The first control is a roleTemplate associated with the service. The roleTemplate identifies specific actions (e.g. read, replace) that can be performed against a certain scope (resource or set of resources).
of a first entity external to said operating environment,	The first entity is the administrator of the server database, or other entity with authority over its content that sets up the roleTemplates and scopes. That entity is independent from and located remotely from the secure server.
and securely receives a second control	A role element specified by a specific end user, which is securely received by the secure server using the SOAP-SEC protocol.
of a second entity external to said operating environment, said second entity being different from said first entity;	The end user is located remotely from the secure server.
and a protected processing environment, operatively connected to said communications arrangement, that:	The protected processing environment is the .NET security service (authorization system) operating within the server. The server uses the SOAP-SEC communication protocol to receive controls.
(a) securely processes, using at least one resource, a data item logically associated with said first and second controls, and	"Securely processes" is performing the requested operation on secure server running .NET. The system will perform the requested operation ensuring that the user has no access to information outside the

	scope computed. The resource is the server software and/or hardware used to process the two controls and user data. The first control is the roleTemplate for the service. The second control is the role element for an individual user. The data item is the end user's stored content (e.g. calendar, email inbox, etc.).
(b) securely applies said first and second controls to manage said resource for controlling use of said data item.	The secure server determines the result scope (visible node set) for the operation that is computed from the role element and the roleTemplate. That result scope is used to manage the data item.
64. A system as in claim 36 wherein said communications arrangement receives said first and second controls from at least one remote location over at least one telecommunications link.	The remote location is the site where the user's or administrator's application is running. The telecommunication link can be the Internet, intranet, VPN or other similar channels.
75. A system as in claim 36 wherein said protected processing environment combines said first and second controls to provide a combined control arrangement.	The role scope incorporating the role element and the role Template.
82. A system as in claim 36 wherein said communications arrangement independently receives said first and second controls at different times.	Administrator and user controls will ordinarily be received at different times.
95. A secure operating environment system as in claim 36 wherein said communications arrangement also receives a data item separately and at a different time from at least one of said first control and said second control.	This is the normal case for .NET My Services. The user's content is normally stored and updated independently of the setting of scope elements, role elements and roleTemplates.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,157,721

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
	Product Infringing: Windows CE for Automotive
1. A security method comprising:	**WCEfA** is Microsoft Windows CE for Automotive, sometimes also known by its former name, AutoPC 2.0. With WCEfA an OEM can assign their device to a class that only accepts certain kinds of software. The device can be set to accept 1) any software with the correct processor/version 2) only certified software or 3) only software from the OEM or Microsoft. These Security (or Trust) levels also control to which kernel APIs and middleware APIs the software has access. Background: "Microsoft Software Install Manager (SIM), a component of WCEfA, allows you to control what can be installed on your device platform. You can define your platform as being open, closed or restricted to new installations, and SIM will enforce these designations." (D,pg.1) "Anything can be installed on an open platform, as long as the applications are compiled for the appropriate processor. At the other extreme, no third-party software can be installed on a closed platform. Only certified applications can be installed on a restricted platform." (D, pg.1) "By restricting installations to compliant applications, the risk of installing and using incompatible or harmful software is greatly reduced, while still keeping the device open for robust, quality applications that enhance the user experience." (F, pg.1) WCEfA also has a Security Layer whose purpose is to "Create an abstraction layer of security surrounding ISV applications to limit and/or deny access to key Windows CE kernel API calls and WCEfA middleware APIs." I, pg. 1)
(a) digitally signing a first load module with a first digital signature designating the first load module for use by a first device class;	A *first load module* is a WCEfA software component in a signed .PE file. The *first device class* is a device that only allows software designated as "restricted" (or higher) to be installed. "Restricted" software is software that has been certified. With restricted software, the device also implements a Security Layer functionality that limits the kernel and WCEfA API calls that the software can make.

"SIM Level: 1 = Restricted
Description: Only properly certified CEI (WCEfA device installation) files can be installed on the device. Remote execution is restricted to executables with master key.
Key: Logo certified CEI file required. CEI files or EXEs with master keys permitted." (F, pg.1)

"The kernel loader calls it each time a module is loaded by Windows CE. It returns one of the following values that determine the module's access to kernel resources:

Value
Meaning

OEM_CERTIFY_TRUST (2)
The module is trusted by the OEM to perform any operation.

OEM_CERTIFY_RUN (1)
The module is trusted by the OEM to run but is restricted from making certain function calls.

OEM_CERTIFY_FALSE (0)
The module is not allowed to run.

" (H, pg. 1)

Digitally signing: "Before the kernel loads a file, it uses the OEMCertifyModule function to verify that the file contains the proper signature." (N, pg.1)

"Signfile.exe: This tool signs an executable with a supplied private key. You can use the following command parameters with this tool....-s AttribString, specifies an optional attribute string to be included in the signature. For example, you could add a string to indicate the trust level of the application." (O. Pg. 1)

In the MSDN article Verifying the Signature, the sample code segment states
"//the file has a valid signature
// we expect the trust level to be returned as signed data...
//case 'R' : dwTrustLevel = OEM_CERTIFY_RUN" (N, pg.2)

"The WCEfA Security Layer isolates installed applications from making unrestricted kernel and WCEfA API calls. This allows the OEM to assign one of three levels of security to applications and drivers installed in RAM when they are loaded into the system. The three levels are Trusted...,Restricted..., and Blocked...On the systems level, the WCEfA Security

	layer fits between ISV applications and isolates these software modules from having free access to all WinCE kernel calls and WCEfA middleware APIs." (I, pg. 1) The developer submits their application for certification. If it passes, then the .cei file (a form of cab file) receives a certification key from the certifier. The signed PE is within this .cei file.

(b) digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class;	A *second load module* is a WCEfA software component is a signed PE file. The *second device class* with a different tamper resistance or security level is a device that is "Closed", that is, it will not allow third party to software to be installed. A closed device only allows trusted software to run. The Security Layer setting of "Trusted" allows the Microsoft and OEM software full access to kernel and middleware APIs.
	
	In the MSDN article Verifying the Signature, the sample code segment states
	"//the file has a valid signature
	// we expect the trust level to be returned as signed data...
	//case 'T' : dwTrustLevel = OEM_CERTIFY_TRUST" (N, pg.2)
	
	"Signfile.exe: This tool signs an executable with a supplied private key. You can use the following command parameters with this tool....-s AttribString, specifies an optional attribute string to be included in the signature. For example, you could add a string to indicate the trust level of the application. (O. Pg. 1)
	
	"SIM Level: 2 = Closed
	Description: Platform is limited to software supplied directly by OEM or Microsoft. Third-party applications cannot be installed. ...
	Key: Master key required for any install or remote execution." (F, pg.1)
	
	Related to the Security Layer, the Trusted level "is most likely reserved for MS and OEM applications and drivers." (I, pg. 1)
	
	Whereas the .cei files for certified software have a certification key (sometimes call MS Logo key), the .cei files from Microsoft or the OEM have a master key attached. ""Master key required for any install or remote execution." (F, p.g1)

(c) distributing the first load module for use by at least one device in the first device class; and	*First load module* is the certified software from a third party that will be run as part of the "Restricted" *first device class*. "Once your application is complete, send the .cei file to

	the organization that is performing validation or certification for the OEM. They would validate it, then either reject or return a .cei that has been stamped with a certification key. You would then reproduce this .cei file on CD-ROM or a compact flash card and distribute." (D, p.g 5) "APCLoad compares the device SIM level against the .cei file certification key, and either allows the installation to proceed or prohibits it based on the outcome of this comparison." (D, pg. 2) "Security:. To achieve a high level of reliability, WCEfA is carefully designed to: - Control the installation of certified and tested software and drivers. - Limit the access of system services by installed module. - Monitor the proper execution of software..." (G, pg. 1)
(d) distributing the second load module for use by at least one device in the second device class.	The *second load module* is the certified software from the OEM or Microsoft that will be run as part of the "Closed" *second device class.* "You may need to change ROM components after your device ships, either to fix a problem, or to provide enhanced functionality. For this purpose, the OEM is given a CEIBuild that adds a master key to a .cei file. CEI files stamped with this master key can be installed on an open, closed or a restricted platform." (D, pg. 3) "Trusted: The application is registered as a completely trusted module and allowed full access to the kernel APIs and WCEfA APIs. This mode is mostly likely reserved for MS and OEM applications and drivers. Note that applications and drivers included in ROM are automatically given trusted status." (I, pg.1)
References: [D] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnceauto/html/WinCAuto_SIM.asp [F] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/apcguide/htm/ceibuildrev_8.asp [G] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/apcguide/htm/securityrev.asp [H] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/apcguide/htm/securityrev_7.asp [I] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/apcguide/htm/reliabilityrev_3.asp [N] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcedsn40/htm/cgconVerifyingSignature.asp [O] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wceoem/htm/os_secur_6.asp	

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,157,721

5.	Product infringing: Windows Hardware Quality Lab certification services, and operating system products that support driver signature technology.
A software verifying method comprising:	Microsoft encourages manufacturers to have their device drivers tested and signed. For example, only signed drivers will ship "in-the-box." Also, Microsoft's driver ranking prefers signed drivers to unsigned drivers. Microsoft Web Page – Can't Find a Test Category for Your Driver? WHQL's long-term objective is to be able to digitally sign all drivers. Although we do not currently have test programs for certain driver types, such as specialized device drivers and software filter drivers, WHQL is investigating a long term solution to expand the categories of drivers tested under Windows 2000 and ultimately all Windows operating systems. We are already formulating a test program for anti-virus file system filters, and plan to address other file system filter drivers as soon as the initial program is in place.
(a) testing a load module	The driver will be tested for each version of the operating system it supports and against the device class specification that apply to the device's class. The driver package is a load module. A driver package contains one or more of the following files: A device setup information file (INF file) A driver catalog (.cat) file One or more optional co-installers Microsoft operates the Window Hardware Quality Lab, which tests drivers submitted by driver manufactures. The manufacturer can test their own driver using the Microsoft testing kit and submit the test results to WHQL when requesting a signature. Additionally, Microsoft or a testing facility working with Microsoft can perform the testing.
having at least one specification associated	The manufacturer-written INF file, which

therewith,	is part of the driver package, is a specification. Microsoft Windows drivers must have an INF file in order to be installed.
the specification describing one or more functions performed by the load module;	The INF Version section specifies its device class. One use of the device class is to identify the specific Windows compatibility specification that relate to the device class. These specifications will vary by device class in part because the function of each device can vary among class. The INF incorporates by reference the Microsoft supplied device class-specific specification by identifying its class in the INF. The INF can include operating system "decorating" to specify the operating system architecture, major and minor version, product and suite the driver is intended for and can further use this decorating to specify what operating systems for which it is not intended. Because the functionality of each of the operating systems may vary the driver must be tested for each applicable operating system. Qualification Service Policy Guide – Hardware Category Policies You must select the correct hardware category for your device. If you select the wrong hardware category for your device, your submission will fail. For example, if you have a storage/hard drive device, but you select storage/tape drive as your hardware category, your submission will fail. Windows XP HCT 10.0 Q & A – Windows XP Logos **Q**: Which "Designed for Windows XP" logos are available for my product? **A**: Devices and systems qualify for a "Designed for Windows" logo after passing testing with the appropriate WHQL test kit on all operating systems specified by the logo. "Designed for Windows" Logos for Device and System Programs lists which logos are available for each type of product.
(b) verifying that the load module satisfies the specification; and	The Microsoft WindowsXP Hardware Compatibility Test (HCT) kit version 10.0 includes the tests, test documentation, and

	submission processes that are required to participate in the Microsoft Windows Logo Program for Hardware for the Windows XP Professional operating system. To qualify to use the "Designed for Windows" logo for hardware, products must pass testing with the Microsoft Windows HCT kit. The HCT kits are organized by hardware type. As mentioned above, the manufacturer can test their own driver using the Microsoft testing kit and submit the test results to WHQL when requesting a signature. Additionally, Microsoft or a testing facility working with Microsoft can perform the testing.
(c) issuing at least one digital certificate attesting to the results of the verifying step.	When a driver package passes WHQL testing, WHQL generates a separate CAT file containing a hash of the driver binaries and other relevant information. WHQL then digitally signs the CAT file using Digital Signature cryptographic technology and sends it to the vendor. Driver signing does not change the driver binaries or the INF file submitted for testing. Microsoft uses digital signatures for device drivers to let users know that drivers are compatible with Microsoft Windows XP, Windows 2000, and Windows Me. A driver's digital signature indicates that the driver was tested with Windows for compatibility and has not been altered since testing.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,157,721

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
14.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A first protected processing environment comprising:	A personal computer running Windows XP, Windows 2000, or Windows 2003
a first tamper resistant barrier having a first security level, and	The tamper resistant barrier is the Office 2003 IRM client environment and includes the signed digital certificate identifying the user. If the certificate is tampered with, or if certain, sensitive IRM processes or modules are debugged or tampered with, the system will cease to operate. The first security level is the "Security Level" which has been selected for a particular Office Application, e.g., Word.
at least one arrangement within the first tamper resistant barrier that prevents the first protected processing environment from executing the same load module accessed by a second protected processing environment having a second tamper resistant barrier with a second security level different from the first security level.	The arrangement that prevents a load module from running in one PPE and not in another is the type and characteristics of a particular Load Module (VBA program within a document or add-in); i.e., signed, script author, code capabilities, etc., and the "Security Level" settings.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,157,721

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
18.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A method for protecting a first computing arrangement surrounded by a first tamper resistant barrier having a first security level, the method including:	The first computing arrangement with a tamper resistant barrier is the Office 2003 IRM client environment and includes the signed digital certificate identifying the user. If the certificate is tampered with, or if certain, sensitive IRM processes or modules are debugged or tampered with, the system will cease to operate. The computing arrangement is being protected from; for example, viruses and malicious code. The first security level is the "Security Level" which has been selected for a particular Office Application, e.g., Word.
preventing the first computing arrangement from using the same software module accessible by a second computing arrangement having a second tamper resistant barrier with a second security level different from the first security level.	The arrangement that prevents a load module from running in one computing arrangement and not in another is the type and characteristics of a particular software module (VBA program within a document or add-in); i.e., signed, script author, code capabilities, etc., and the "Security Level" settings.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,157,721

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
34.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A protected processing environment comprising:	A personal computer running Windows XP, Windows 2000, or Windows 2003
a first tamper resistant barrier having a first security level,	The first tamper resistant barrier is the Office 2003 IRM client environment and includes the signed digital certificate identifying the user. If the certificate is tampered with, or if certain, sensitive IRM processes or modules are debugged or tampered with, the system will cease to operate. The first security level is the "Security Level" which has been selected for a particular Office Application, e.g., Word.
a first secure execution space, and	The secure execution space is process space allocated by the operating system for the Microsoft Office host application to run. This host application (e.g., Word) executes the VBA code within this process space. This execution space (application) is secure because the IRM environment takes steps to insure that it is "trusted", the application is signed, and the document which includes the VBA code is protected by IRM policy and then encrypted and signed.
at least one arrangement within the first tamper resistant barrier that prevents the first secure execution space from executing the same executable accessed by a second secure execution space having a second tamper resistant barrier with a second security level different from the first security level.	The arrangement that prevents a load module from running in one computing arrangement and not in another is the type and characteristics of a particular software module (VBA program within a document or add-in); i.e., signed, script author, code capabilities, etc., and the "Security Level" settings.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,157,721

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
34.	Product Infringing: Microsoft Common Language Runtime and ASP.NET
A protected processing environment comprising:	Microsoft Common Language Runtime and ASP.NET
a first tamper resistant barrier having a first security level,	TAMPER RESISTANT BARRIER The first tamper resistant barrier is the application domain in the CLR. The runtime hashes the contents of each file loaded into the application domain and compares it with the hash value in the manifest. If two hashes don't match, the assembly fails to load.[1] Also "*Code running in one application cannot directly access code or resources from another application. The common language runtime enforces this isolation by preventing direct calls between objects in different application domains. Objects that pass between domains are either copied or accessed by proxy.*"[2] SECURITY LEVELS The security levels of the application domain if different by setting the trust level assigned to an outside application using the "trust" element in the web.config for the ASP.NET application. Syntax- <trust level="Full/High/Low/None" originUrl="url"/> Example- <trust level="High" originUrl=http://www.SomeOtherCompany.com/default.aspx /> [7]
a first secure execution space, and	The application domain is the execution space for a particular application.
at least one arrangement within the first tamper resistant barrier that prevents the first secure execution space from executing the same executable accessed by a second secure execution space having a second tamper resistant barrier with a second security level different from the first security level.	The second secure execution space is another application domain that has a different trust level for an outside application. If second app domain gives Full trust to the outside application; whereas the first one doesn't, the first app domain won't be able to execute the application that requires full trust permission.
	References: [1]

	www.microsoft.com/germany/ms/msdnbiblio/dotnetrk/doc/assembly.doc [2] msdn.Microsoft.com/library/en-us/cpguide/html/cpconapplicationdomainsoverview.asp?frame=true [7] LaMacchia,etc, .NET Framework Security, Addision-Wesley, 2002

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,157,721

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
34.	Product Infringing: Products containing Microsoft Common Language Runtime or Compact Common Language Runtime and products implementing the Common Language Infrastructure specification.
A protected processing environment comprising:	Microsoft Common Language Runtime and .NET Framework SDK:
a first tamper resistant barrier having a first security level,	TAMPER RESISTANT BARRIER The first tamper resistant barrier is the application domain in the CLR. The runtime hashes the contents of each file loaded into the application domain and compares it with the hash value in the manifest. If two hashes don't match, the assembly fails to load. [1] Also "*Code running in one application cannot directly access code or resources from another application. The common language runtime enforces this isolation by preventing direct calls between objects in different application domains. Objects that pass between domains are either copied or accessed by proxy.*"[2] SECURITY LEVELS Application domains have different security levels by setting security policy of the application domain programmatically. [3] "*It has different security based on code-based security model of .NET. Administrators and hosts use code-access security to decide what code can do, based on characteristics of the code itself, regardless of what user is executing the code. The code characteristics are called* evidence *and can include the Web site or zone from which the code was downloaded, or the digital signature of the vendor who published the code.*" "*When the security manager needs to determine the set of permissions that an assembly is granted by security policy, it starts with the enterprise policy level. Supplying the assembly evidence to this policy level will result in the set of permissions granted from that policy level. The security manager typically continues to collect the permission sets of the policy levels below the enterprise policy* [including the app domain] *in the same*

	fashion. These permission sets are then intersected to generate the policy system permission set for the assembly. All levels must allow a specific permission before it can make it into the granted permission set for the assembly." Example of granted permission sets from a policy – *Condition: All code, Permission Set: Nothing* *Condition: Zone: Internet, Permission Set: Internet Condition: URL: www.monash.edu.au, Permission Set: MonashPSet* *Condition: Strong Name: m-Commerce, Permission Set: m-CommercePSet [4]* Another difference in security levels can be whether the verification process is turned off or on, "*Managed code must be passed through a verification process before it can be run (unless the administrator has granted permission to skip the verification). The verification process determines whether the code can attempt to access invalid memory addresses or perform some other action that could cause the process in which it is running to fail to operate properly. Code that passes the verification test is said to be type-safe. The ability to verify code as type-safe enables the common language runtime to provide as great a level of isolation as the process boundary, at a much lower performance cost.*" [5]
a first secure execution space, and	The application domain is the execution space for a particular application.
at least one arrangement within the first tamper resistant barrier that prevents the first secure execution space from executing the same executable accessed by a second secure execution space having a second tamper resistant barrier with a second security level different from the first security level.	The second secure execution space is another application domain that has a different security policy than the first. If second app domain's security policy doesn't give any permission to code from internet zone, but first app domain does, then the code would run in first app domain and not in second.[6]
	References: [1] www.microsoft.com/germany/ms/msdnbiblio/dotnetrk/doc/assembly.doc [2] msdn.Microsoft.com/library/en-us/cpguide/html/cpconapplicationdomainsoverview.asp?frame=true

	[3] LaMacchia,etc, .NET Framework Security, Addision-Wesley, 2002, p.113 [4] Watkins, Demien, "An Overview of Security in the .NET Framework", from MSDN Library, January 2002 [5] same as [2] [6] msdn.Microsoft.com/library/en-us/cpguide/html/cpconapplicationdomainlevelsecuritypolicy.asp?frame=true

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,157,721

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
38.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A method for protecting a first computing arrangement surrounded by a first tamper resistant barrier having a first security level, the method including:	The first computing arrangement surrounded by a tamper resistant barrier is the Office 2003 IRM client environment and includes the signed digital certificate identifying the user. If the certificate is tampered with, or if certain, sensitive IRM processes or modules are debugged or tampered with, the system will cease to operate. The first security level is the "Security Level" which has been selected for a particular Office Application, e.g., Word.
preventing the first computing arrangement from using the same software module accessed by a second computing arrangement having a second tamper resistant barrier with a second security level different from the first security level.	The computing arrangement that prevents a software module from running in one computing arrangement and not in another is the type and characteristics of the particular software module (VBA program within a document or add-in); i.e., signed, script author, code capabilities, etc., and the "Security Level" settings.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
2.	Product Infringing: Windows Media Rights Manager and Windows Media Player
A system including:	
(a) a first apparatus including,	Consumer's computer, as shown in WMRM SDK
(1) user controls,	Consumer's computer, as shown in WMRM SDK
(2) a communications port,	Consumer's computer, as shown in WMRM SDK
(3) a processor,	Consumer's computer, as shown in WMRM SDK
(4) a memory storing:	Consumer's computer, as shown in WMRM SDK
(i) a first secure container containing a governed item, the first secure container governed item being at least in part encrypted; the first secure container having been received from a second apparatus;	Secure container (packaged Windows Media file), received by consumer's computer from "Content provider" (WMRM SDK, Step 3), which contains encrypted governed item ("Encrypted content")
(ii) a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item, the first secure container rule [sic], the first secure container rule having been received from a third apparatus different from said second apparatus; and	Rights portion of signed license, received by consumer's computer from "License issuer" (WMRM SDK, Step 9)
(5) hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	Windows Media Player and Windows Media Rights Manager
(6) a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus, said protected processing environment including hardware or software used for applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	1st and 2nd rules consist of any two valid rules as specified in the Window Media Rights Manager SDK; protected processing environment includes Windows Media Rights Manager and Windows processes for protecting operation of Windows Media Rights Manager. Licenses can be used to convey multiple rules.
(7) hardware or software used for	Any hardware or software employed in

transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	transmitting Windows Media files, including for example consumer's computer's communication port and Windows Media Player (WMRM SDK, Step 3)

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
2.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A system including:	
a first apparatus including, user controls, a communications port, a processor, a memory storing:	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive or RAM.
a first secure container containing a governed item, the first secure container governed item being at least in part encrypted; the first secure container having been received from a second apparatus;	An encrypted IRM-governed email received from a remote computer. The encrypted IRM-governed email contains an encrypted IRM-governed email message.
a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item, the first secure container rule, the first secure container rule having been received from a third apparatus different from said second apparatus; and	The first secure container rule is received from the RMS server in the form of a use license. This use license contains rules generated by the RMS server specifically for the user (or user's group)
hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The RM-enabled device contains hardware or software for receiving and opening secure emails. The secure email has the capacity to contain an IRM-governed email message, with a rule being associated with each email. The rules associated with the secure emails are rules that come as part of the original email as well as rules that come back from the RMS.
a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus, said protected processing environment including hardware or software used for applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of	Protected information on the RM-enabled device is protected by the use of at least cryptographic techniques. The rule governing the email works together with an additional rule to determine what access to or use (if any) are allowed with respect to the IRM-governed email message. For example, the additional rule may be

access to or use of a governed item contained in a secure container; and	received together with the rule in the use license.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	The device includes hardware or software used for transmitting or receiving secure emails. For example, RM-enabled OUTLOOK is designed to transmit and receive encrypted IRM-governed emails to/from other devices.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
2.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A system including:	
a first apparatus including, user controls, a communications port, a processor, a memory storing:	A device with user controls, a communications port, a processor, and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive or RAM.
a first secure container containing a governed item, the first secure container governed item being at least in part encrypted; the first secure container having been received from a second apparatus;	The first secure container is an encrypted IRM-protected document. This encrypted IRM-governed document is, for example, received from a remote computer, as an attachment to an IRM-governed email or downloaded from a document server or web site.
a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item, the first secure container rule, the first secure container rule having been received from a third apparatus different from said second apparatus; and	The first secure container rule is received from the RMS server in the form of a use license. This use license contains rules generated by the RMS server specifically for the user (or user's group).
hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The RM-enabled device contains hardware or software for receiving and opening secure documents. The secure documents have the capacity to contain IRM-governed content, with a rule being associated with each secure document. The rules associated with said secure documents are the rules that come as part of the originally received document as well as rules that come back from the RMS server.
a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus,	Protected information on the RM-enabled device is protected by the use of at least cryptographic technique. The rule governing the document works

said protected processing environment including hardware or software used for applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	together with an additional rule to determine what access to or use (if any) are allowed with respect to the IRM-governed document. For example, the additional rule may be associated with an email to which the document was attached, or received together with the rule in the use license.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	The device includes hardware or software used for transmitting or receiving secure documents. For example, RM-enabled OUTLOOK is designed to transmit and receive to/from other devices emails with IRM-governed documents attached thereto.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
3.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A system including:	
a first apparatus including, user controls, a communications port, a processor, a memory storing:	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive or RAM.
a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	The first secure container containing a governed item is an IRM protected email. Both the email and attachment are IRM protected, each having their own rules, each being encrypted.
a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item; and	The rule governing the email (a first secure container rule) governs said first secure container governed item.
a second secure container containing a digital certificate;	The second secure container is the IRM protected attachment's derived license request object. The license request object contains the Publishing license and a signed digital certificate.
hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The RM (IRM) enabled computer has software for receiving and opening secure containers. The IRM secure containers have capacity to contain a governed item, with a secure container rule being associated with each of said secure containers.
a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus, said protected processing environment including hardware or software used for	Protected information on the RM-enabled computer is protected by the use of at least cryptographic techniques. The rules governing the email itself (first

applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	secure container rule) and the rules governing the attachment work together to determine what access to or use (if any) will be allowed with respect to the governed item.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	IRM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured containers to/from other computers.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
3.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A system including:	
a first apparatus including, user controls, a communications port, a processor, a memory storing:	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive or RAM.
a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	The first secure container containing a governed item is an IRM protected document, which is an attachment within an IRM protected email message. The governed item is the document's content. Both the email message and attachment are encrypted and have associated usage rules due to IRM protection.
a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item; and	A use license for the IRM protected document specifies rules governing access to or use of said first secure container governed item.
a second secure container containing a digital certificate;	The second secure container is the IRM protected email message. The IRM protected attachment includes a publishing license and an owner certificate, both of which are signed XrML digital certificates. The attachment (including embedded certificates) is contained within the IRM protected email message (said second secure container).
hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The RM (IRM) enabled computer has software for receiving and opening secure containers. The IRM secure containers have capacity to contain a governed item, with a secure container rule being associated with each of said secure containers.
a protected processing environment at least in part protecting information contained in said protected processing environment from	Protected information on the RM-enabled computer is protected by the use of at least cryptographic techniques.

tampering by a user of said first apparatus, said protected processing environment including hardware or software used for applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	The rules governing the attachment (first secure container rule) and the rules governing the email message (second secure container rule) work together to determine what access to or use (if any) will be allowed with respect to the governed item.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	RM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured containers to/from other computers.
4. A system as in claim 3,	
said memory storing a rule associated with said second secure container, said rule associated with said second secure container at least in part governing at least one aspect of access to or use of said digital certificate.	All parts of the attachment (including embedded signed XrML licenses/certificates) are protected by the enclosing email message and governed by the associated email rules (second secure container rule).

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A system including:	
a first apparatus including, user controls, a communications port, a processor, a memory storing:	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive or RAM.
a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	first secure container containing a governed item is an IRM protected email. Both the email and attachment are IRM protected, each having their own rules, each being encrypted.
a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item; and	The rule governing the email (a first secure container rule) governs said first secure container governed item.
a second secure container containing a digital signature, the second secure container being different from said first secure container;	The second secure container is the IRM protected attachment's derived license request object. The license request object contains the Publishing license and a signed digital certificate.
hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The RM (IRM) enabled computer has software for receiving and opening secure containers. The IRM secure containers have capacity to contain a governed item, with a secure container rule being associated with each of said secure containers.
a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus, said protected processing environment including hardware or software used for applying said first secure container rule and a	Protected information on the RM-enabled computer is protected by the use of at least cryptographic techniques. The rules governing the email itself (first secure container rule) and the rules governing

second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	the attachment will work together to determine what access to or use (if any) will be allowed with respect to the governed item.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	RM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured containers to/from other computers.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A system including:	
a first apparatus including, user controls, a communications port, a processor, a memory storing:	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive or RAM.
a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	first secure container containing a governed item is an IRM protected email. Both the email and attachment are IRM protected, each having their own rules, each being encrypted.
a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item; and	The rule governing the email (a first secure container rule) governs said first secure container governed item.
a second secure container containing a digital signature, the second secure container being different from said first secure container;	The second secure container is the IRM email attachment. This attachment and its publishing license are signed.
hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The RM (IRM) enabled computer has software for receiving and opening secure containers. The IRM secure containers have capacity to contain a governed item, with a secure container rule being associated with each of said secure containers.
a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus, said protected processing environment including hardware or software used for applying said first secure container rule and a	Protected information on the RM-enabled computer is protected by the use of at least cryptographic techniques. The rules governing the email itself (first secure container rule) and the rules governing

second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	the attachment work together to determine what access to or use (if any) will be allowed with respect to the governed item.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	RM-enabled applications, e.g., OUTLOOK,are designed to transmit and receive RM secured containers to/from other computers.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A system including:	
a first apparatus including, user controls, a communications port, a processor, a memory storing:	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive or RAM.
a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	The first secure container containing a governed item is an IRM protected document, which is an attachment within an IRM protected email message. The governed item is the document's content. Both the email message and attachment are encrypted and have associated usage rules due to IRM protection.
a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item; and	A use license for the IRM protected document specifies rules governing access to or use of said first secure container governed item.
a second secure container containing a digital signature, the second secure container being different from said first secure container;	The second secure container is the IRM protected email message. The IRM protected attachment includes a publishing license and an owner certificate, both of which are signed XrML digital certificates. The attachment (including embedded certificates) is contained within the IRM protected email message (said second secure container).
hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The RM (IRM) enabled computer has software for receiving and opening secure containers. The IRM secure containers have capacity to contain a governed item, with a secure container rule being associated with each of said secure containers.
a protected processing environment at least in part protecting information contained in said	Protected information on the RM-enabled computer is protected by the use of at least

protected processing environment from tampering by a user of said first apparatus, said protected processing environment including hardware or software used for applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	cryptographic techniques. The rules governing the attachment (first secure container rule) and the rules governing the email message (second secure container rule) work together to determine what access to or use (if any) will be allowed with respect to the governed item.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	RM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured containers to/from other computers.
6. A system as in claim 5,	
said memory storing a rule at least in part governing an aspect of access to or use of said digital signature.	All parts of the attachment (including embedded signed XrML licenses/certificates) are protected by the enclosing email message and governed by the associated email rules (second secure container rule).

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
28.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A system including:	
a first apparatus including; user controls, a communications port, a processor, a memory containing a first rule,	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive or RAM. The first rule governs use of an IRM protected document (e.g., an IRM rule permitting a document to be read by specified users or barring access to IRM-governed information from specified users, applications, or other principals).
hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The RM-enabled device contains hardware or software for receiving and opening secure containers. The secure email has the capacity to contain an IRM-governed email message, with a rule being associated with each email.
a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus, said protected processing environment including hardware or software used for applying said first rule and a secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item; and	Protected information on the RM-enabled device is protected by the use of at least cryptographic techniques. The secure container rule is an IRM rule governing access to the IRM protected document (e.g., a rule permitting editing by specified users). The rule governing the email works together with an additional rule to determine what access to or use (if any) are allowed with respect to the IRM-governed email message (the document's content). For example, the additional rule may be received together with the rule in the use license, may be associated with a publishing license, may be associated with user certification, revocation lists, or exclusion policies, or may be received from any other source.
hardware or software used for transmission of	The device includes hardware or software used

secure containers to other apparatuses or for the receipt of secure containers from other apparatuses; and	for transmitting or receiving secure containers. For example, RM-enabled OUTLOOK is designed to transmit and receive encrypted IRM-governed emails to/from other devices.
a second apparatus including:	
user controls, a communications port, a processor, a memory containing a second rule,	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive or RAM. The second rule governs use of an IRM protected document (e.g., an IRM rule permitting a document to be read by specified users or barring access to IRM-governed information from specified users, applications, or other principals).
hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The RM-enabled device contains hardware or software for receiving and opening secure containers. The secure email has the capacity to contain an IRM-governed email item, with a rule being associated with each secure containers.
a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said apparatus, said protected processing environment including hardware or software used for applying said second rule and a secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item;	Protected information on the RM-enabled device is protected by the use of at least cryptographic technique. The secure container rule is an IRM rule governing access to the IRM protected document (e.g., a rule permitting editing by specified users). The rule governing the email works together with an additional rule to determine what access to or use (if any) are allowed with respect to the IRM-governed item (the document's content). For example, the additional rule may be received together with the rule in the use license, may be associated with a publishing license, may be associated with user certification, revocation lists, or exclusion policies, or may be received from any other source.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses; and	The device includes hardware or software used for transmitting or receiving secure containers. For example, RM-enabled OUTLOOK is designed to transmit and receive encrypted IRM-governed emails to/from other devices.
an electronic intermediary, said intermediary including a user rights authority clearinghouse.	The RMS Server (Microsoft hosted or otherwise) constructs a 'use license' specific to a piece content and targets it to a specific user.

29. A system as in claim 28, said user rights authority clearinghouse operatively connected to make rights available to users.	The RMS server sends *use licenses* to users through a communications port, e.g., Ethernet, serial, satellite, "the internet" These use licenses include rights. The clearing functionality of the RMS is operatively connected to the RMS server.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

28.	Product Infringing: Windows Media Rights Manager and Windows Media Player
A system including:	
(a) a first apparatus including;	Consumer's computer, as shown in WMRM SDK
(1) user controls,	Consumer's computer, as shown in WMRM SDK
(2) a communications port,	Consumer's computer, as shown in WMRM SDK
(3) a processor,	Consumer's computer, as shown in WMRM SDK
(4) a memory containing a first rule,	Memory is in the consumer's computer, first rule is a right received as part of a signed license (WMRM SDK, Step 9)
(5) hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	Consumer's computer receives Windows Media file (secure container) via communications port (WMRM SDK, Step 3) and applies secure container rule or rules via Windows Media Player and Windows Media Rights Manager.
(6) a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus, said protected processing environment including hardware or software used for applying said first rule and a secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item; and	Processing environment includes Windows Media Rights Manager and Windows processes for protecting operation of Windows Media Rights Manager
(7) hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses; and	Hardware or software employed in transmitting Windows Media files, including for example consumer's computer's communication port and Windows Media Player (WMRM SDK, Step 3)
(b) a second apparatus including:	2nd consumer's computer
(1) user controls,	2nd consumer's computer
(2) a communications port,	2nd consumer's computer
(3) a processor,	2nd consumer's computer
(4) a memory containing a second rule,	Memory is in the 2nd consumer's computer, first rule is a Right received as part of a signed license (WMRM SDK, Step 9)
(5) hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to contain	2nd consumer's computer receives Windows Media file (secure container) via communications port (WMRM SDK, Step 3) and applies secure container rule or rules via

a governed item, a secure container rule being associated with each of said secure containers;	Windows Media Player and Windows Media Rights Manager.
(6) a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said apparatus, said protected processing environment including hardware or software used for applying said second rule and a secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item;	Processing environment includes Windows Media Rights Manager and Windows processes for protecting operation of Windows Media Rights Manager; processing environment applies multiple rules in combination
(7) hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses; and	Hardware or software employed in transmitting Windows Media files, including for example 2nd consumer's computer's communication port and Windows Media Player (WMRM SDK, Step 3)
(c) an electronic intermediary, said intermediary including a user rights authority clearinghouse.	License Issuer
29. A system as in claim 28,	
said user rights authority clearinghouse operatively connected to make rights available to users.	License Issuer, operatively connected to consumer's computer (WMRM SDK, Step 9)

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
56.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A method of securely delivering an item, including the following steps:	
performing an authentication step;	The RM-enabled application, e.g., Word, OUTLOOK, PowerPoint, etc., must be authenticated before it is allowed access to or use of the content.
associating a digital signature with said item;	The RM protected content is signed.
incorporating said item into a first secure electronic container, said item being at least in part encrypted while in said container, said incorporation occurring in an apparatus containing a first protected processing environment, said protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said apparatus;	RM-protected content is packaged with rules and encrypted. Protected information on the RM enabled computer is protected by the use of at least cryptographic techniques.
in said protected processing environment, associating a first rule with said first secure electronic container, said first rule at least in part governing at least one aspect of access to or use of said item;	The IRM-protected document (said item) has an associated rule or rules.
authenticating an intended recipient of said item;	A recipient of IRM-protected content must be authenticated before being allowed access to or use of the content.
transmitting said first secure electronic container and said first rule to said intended recipient; and	The document is sent via IRM-protected email as an attachment.
using a second protected processing environment, providing said intended recipient access to at least a portion of said item, said access being governed at least in part by said first rule and by a second rule present at said intended recipient's site.	The email is received at another IRM-enabled computer. The first said rule is the rule(s) associated with the attached document, and the second rule is the rule(s) received that govern the email itself.

126.	Product Infringing: Windows Hardware Quality Labs Authentication services, Windows operating Systems (such as Windows XP) that support the driver signing features, and any product using Driver Signing feature
A method of providing trusted intermediary services including the following steps:	
at a first apparatus, receiving an item from a second apparatus;	Microsoft's Window Hardware Quality Labs (WHQL) (first apparatus) receiving driver package (item) from independent hardware vendor (IHV) or any driver developer (second apparatus).
associating authentication information with said item;	The signature information of a security catalog file (see next element of claim) names Microsoft as the publisher. WHQL's signature is intended to signify that a driver has complied with Microsoft's Windows compatibility and/or Secure Audio Path (SAP) specifications.
incorporating said item into a secure digital container;	The hashes of the files making up the driver package are included in the signed security catalog file for the driver package. The catalog file makes the driver package a secure digital container.
associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;	Driver developers specify rules in an INF file that govern the installation and/or use of the driver. For example, as specified in the INF, the installation events will vary based on the user's operating system version, which includes architecture, product type and suite. The INF logging rules and can further specify security rules that are evaluated when the driver is used. White Paper – Operating-System Versioning for Drivers under Windows XP Setup selects the [*Models*] section to use based on the following rules: If the INF contains [*Models*] sections for several major or minor operating system version numbers, Setup uses the section with the highest version numbers that are not higher than the operating system version on which the installation is taking place.

If the INF [*Models*] sections that match the operating system version also include product type decorations, product suite decorations, or both, then Setup selects the section that most closely matches the running operating system.

Suppose, for example, Setup is running on Windows XP Professional (which is operating system version 5.1), and it finds the following entry in a [Manufacturer] section:

%FooCorp%=FooMfg, NT, NT.5, NT.5.5, NT....0x80

In this case, Setup will look for a [*Models*] section named **[FooMfg.NT.5]**. Setup will also use the **[FooMfg.NT.5]** section if it is running on a Datacenter version of Windows .NET Server, because a specific major/minor version takes precedence over the product type and suite mask.

For example, to create an INF that is intended for use only on Windows XP, the INF file could contain the following:

[Manufacturer]
"Foo Corp." = FooMfg, NT.5.1, NT.5.2
[FooMfg.NT.5.1]
"Foo Device" = FooDev, *FOO1234

Note the omission of the undecorated [FooMfg] section, as well as the omission of the [FooMfg.NT.5.2] section. This INF file would appear to be "empty" on any operating system other than Windows XP.

Access Control List Rules

XP DDK – Tightening File-Open Security in a Device INF File

For Microsoft Windows 2000 and later, Microsoft tightened file-open security in the class installer INFs for certain device classes, including CDROM, DiskDrive, FDC, FloppyDisk, HDC, and SCSIAdapter.

If you are unsure whether the class installer for your device has tightened security on file opens, you should tighten security by using the device's INF file to assign a value to the **DeviceCharacteristics** value name in the registry. Do this within an *add-*

	registry-section, which is specified using the INF AddReg directive.
transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;	Microsoft, IHV, driver developer or any other party distributing signed driver packages transmitting the driver package to user (third apparatus). Since the driver package includes the INF file, it will include the first rule. The protected processing environment (PPE) is Windows operating system with its pertinent services such as Windows File Protection, signature and cryptographic functions, Plug and Play and Set-up and their related default and modifiable policies. The PPE checks for signatures on driver packages and detects situations when the driver package's signature does not match the driver package. Additionally, the Digital Rights Manager (DRM) components (kernel and client) will contribute to making the third apparatus a PPE when the SAP functionality is invoked. [That is, when SAP is required, an additional signature is checked to verify that the driver is SAP compliant and that it hasn't been tampered with.]
said third apparatus receiving said secure digital container and said first rule;	The end-user receiving the driver package.
said third apparatus checking said authentication information; and	A step in the Plug and Play/Setup driver installation process checks signature at installation. Additionally, the DRM component will check the DRM signature when invoking DRM functionality. White Paper – Driver Signing for Windows During driver installation, Windows compares the hashes contained in the driver's CAT file with the computed hash of the driver binaries to determine whether the binaries have changed since the CAT file was created. If a driver fails the signature check or there is no CAT file, what happens next depends on the driver signing policy in effect on the user's system: If the policy is set to Ignore, the driver installs silently, with no message to the user. If the policy is set to Warn, a message warns the user the driver is unsigned, which means that it has not passed WHQL

	testing and might cause problems. The Warn dialog box gives an administrative user the option to override the warning and install an unsigned driver anyway. If the policy is set to Block, the system displays a message that informs the user that the driver cannot be installed because it is not digitally signed.
said third apparatus performing at least one action on said item, said at least one action being governed, at least in part, by said first rule and by a second rule resident at said third apparatus prior to said receipt of said secure digital container and said first rule, said action governance occurring at least in part in said protected processing environment.	The action would be installing and/or using the driver. For example, installation policies govern the actions (ignore, warn or block) taken based on whether a driver is signed or not and these policies (rule) are resident on the third apparatus. Another rule is the "ranking" of available drivers when selecting a driver to install. This ranking process includes whether a driver is signed or not. Another rule is the security access rules that the class installer that will be used to install the device has. In the case of DRM, the content will have associated rules governing its use in a SAP-complaint environment. These rules (the content license) can be resident at the third apparatus particularly in the case when a user is installing a new (SAP-compliant) device that will render previously acquired content or in the case that acquired content cannot be rendered until the user installs required drivers. For example, when installing: The XP driver ranking process and the modifiable default related to signature state of the driver act as the second rule. The driver will be installed only if the first and second rules validate. Operating-System Versioning for Drivers under Windows XP *Default System Policy for Unsigned Drivers* If the user installs an unsigned driver for a designated device class from disk or from another web site, Windows XP/Windows 2000 displays a warning that the driver is unsigned, thus helping to preserve the integrity of the released system. However, by default, Windows XP/Windows 2000

	does not block installation of unsigned drivers, so vendors can get urgent hot-fixes to customers while waiting for WHQL to test the fix. In Windows XP, the default driver signing policy can be changed through the Hardware tab of the System applet on the Control Panel. A user can change the policy to be more restrictive, but not less restrictive on a per-user basis (that is, a user can change Warn to Block, but not to Ignore). An administrator can change the policy to be either more restrictive or less restrictive for all users on the system by checking "Apply the setting as system default." *Driver Ranking* Under Windows XP, the driver ranking strategy has been modified as follows: If an INF file is unsigned, and if neither the [*Models*] section nor the [*DDInstall*] section is decorated with an NT-specific extension, the INF file is considered "suspect" and its rank is shifted into a higher range (that is, worse) than all hardware and compatible rank matches of INF files for which one (or both) of those criteria are met. The new ranking ranges will now be: 0 – 0xFFF (DRIVER_HARDWAREID_RANK) : "trusted" hardware-ID match 0x1000 – 0x3FFF : "trusted" compatible-ID match 0x8000 – 0x8FFF : "untrusted" hardware-ID match 0x9000 – 0xBFFF : "untrusted" compatible-ID match 0xC000 - 0xCFFF : "untrusted" undecorated hardware-ID match (possibly a Windows 9x-only driver) 0xD000 - 0xFFFF : "untrusted" undecorated compatible-ID match (possibly a Windows 9x-only driver)
127. A method as in claim 126, in which said authentication information at least in part identifies said first apparatus and/or a	The authentication information will identify Microsoft, operator of the first apparatus.

user of said first apparatus.	

126.	Products Infringing: Microsoft Software that includes the Authenticode feature, .NET Framework SDK, Visual Studio, Microsoft technology that supports a digital signature function (such as ActiveX), Windows Installer technology.
A method of providing trusted intermediary services including the following steps:	Infringement is based on use Microsoft ActiveX control, Cabinet file, Microsoft Windows Installer, Authenticode and Software Restriction Policy technologies. For example, a software publisher distributing a signed application that has licensed ActiveX controls embedded within it would practice this method.
at a first apparatus, receiving an item from a second apparatus;	The item is unsigned software such as an ActiveX control or any software packaged in a cabinet file or Microsoft Installer (.msi) file. Within the development environment, multiple software developers (working on a second apparatus) will send their unsigned software to a secure location (first apparatus) containing the entity's private signing key. An example entity would be a software publisher. Source: Deploying ActiveX Controls on the Web with the Internet Component Download **The holder of the digital certificate** Keeping your digital certificate safe is very important. Some firms (including Microsoft) do not keep their signature file on site. The signature is kept with the Certificate Authority and files are sent there for signing.
associating authentication information with said item;	Signing the software associates the software publisher's identify with the software. Source: Packaging ActiveX Controls Signing Cabinet Files A .cab file can be digitally signed like an ActiveX control. A digital signature provides accountability for software developers: The signature associates a software vendor's name with a given file. A signature is applied to a .cab file (or control) using the Microsoft Authenticode®

	technology. The .cab tool set assists software developers in applying digital signatures to .cab files by allowing a developer to allocate space in the .cab file for the signature.
incorporating said item into a secure digital container;	Signing software either directly or within a package (cabinet or .msi file) secures it in a digital container. Alternately, the signed ActiveX control could be placed into a signed cabinet file.
associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;	The first rule would be the licensing support code within the ActiveX control and/or conditional syntax statements when the software is within a signed .msi file. When the software is within a signed cabinet file, the first rule can be a rule contained in the software, as is the case when an ActiveX control is packaged in a signed cabinet file. First rule, in the case of ActiveX: When an application with a licensed ActiveX control is started, an instance of the control usually needs to be created. The application accomplishes this by making a call to CreateInstanceLic and passing the license key embedded in the application as a parameter in the call. The ActiveX control performs a string comparison between the embedded license key and its own copy of the license key. If the keys match, an instance of the control is created and the application can execute normally. Source: Using ActiveX Controls to Automate Your Web Pages Run-time licensing Most ActiveX Controls should support design-time licensing and run-time licensing. (The exception is the control that is distributed free of charge.) Design-time licensing ensures that a developer is building his or her application or Web page with a legally purchased control; run-time licensing ensures that a user is running an application or displaying a Web page that contains a legally purchased control. Design-time licensing is verified by control containers such as Visual Basic, Microsoft Access, or Microsoft Visual InterDev®. Before these containers allow a developer to place a control on a form or Web page,

	they first verify that the control is licensed by the developer or content creator. These containers verify that a control is licensed by calling certain functions in the control: If the license is verified, the developer can add it. Run-time licensing is also an issue for these containers (which are sometimes bundled as part of the final application); the containers again call functions in the control to validate the license that was embedded at design time.
transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;	The third apparatus is a user computer or an application server. The protected processing environment (PPE) is Windows operating system, Internet Explorer (IE) and pertinent operating IE services such as Windows File Protection and security, signature and cryptographic functions related to code signing and related policies. The PPE checks for signatures on software or the software packages and detects situations when the signature does not validate as an indication that tampering may have occurred with the item.
said third apparatus receiving said secure digital container and said first rule;	Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.
said third apparatus checking said authentication information; and	Examine the signature information includes verifying that signature was creating using the private key that corresponds to the public key of the publisher.
said third apparatus performing at least one action on said item, said at least one action being governed, at least in part, by said first rule and by a second rule resident at said third apparatus prior to said receipt of said secure digital container and said first rule, said action governance occurring at least in part in said protected processing environment.	The action would be installation and/or use of the distributed software. The second rule can be software restriction policies resident on the machine, which can be invoked at installation and/or runtime. .NET Framework Security – pg 259 and White Paper – Using Software Restriction Policies in Windows XP and Windows .NET Server to Protect Against Unauthorized Software Software Restriction Polices is a policy-driven technology that allows administrators to set code-identity-based rules that determine whether an application is allowed to execute. (.NET Framework Security – pg 259)

	For example, administrators can set rules for all Windows Installer packages coming from the Internet or Intranet zone. As part of the DLL load mechanisms, Software Restriction Policies is invoked and starts to check its most specific rules. Software Restriction Policies get invoked prior to an .exe being able to run. The four types of rules are – hash, certificate, path, and zone. Note: The hash and certificate rules relate directing to the signature information whereas, the path and zone rules do not.
127. A method as in claim 126, in which said authentication information at least in part identifies said first apparatus and/or a user of said first apparatus.	The software publisher, user of first device, is identified in the authentication information.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,185,683

126.	Product infringing: Visual Studio .NET, .NET Framework SDK, Authenticode, Products that contain the .NET CLR, Compact CLR or CLI.
A method of providing trusted intermediary services including the following steps:	.
at a first apparatus, receiving an item from a second apparatus;	First apparatus is a software build or deployment services computer that has access to signing key. The item may be a program, graphic, media object or other resource, from a developer computer, or archive (second apparatus).
associating authentication information with said item;	Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.
incorporating said item into a secure digital container;	Producing signed, strongly named assembly that contains this assembly and associated attributes.
associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;	Including any security demands (such as members of the Microsoft .NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.
transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;	The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.
said third apparatus receiving said secure digital container and said first rule;	Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.
said third apparatus checking said authentication information; and	The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using the private key that corresponds to the

	public key of the publisher.
said third apparatus performing at least one action on said item, said at least one action being governed, at least in part, by said first rule and by a second rule resident at said third apparatus prior to said receipt of said secure digital container and said first rule, said action governance occurring at least in part in said protected processing environment.	The action is executing code that is the item or using code that renders the item. Action is governed by security demands on code that calls the item or on code that calls code included in the .NET assembly that manages said item. The second rule is the machine, enterprise, user, and application configuration file resident rules. Typically these configuration files will be populated before the arrival of most new assemblies in a virtual distribution environment. This action governance occurs in the protected processing environment of the CLR, CLI and/or compact CLR.

127. A method as in claim 126, in which said authentication information at least in part identifies said first apparatus and/or a user of said first apparatus.	The authentication information will identify the .NET Assembly Class company name and trademark attributes that identify the apparatus or user of the first apparatus as being a member of an entity or a branded source (brand name).

126.	Product infringing: Visual Studio .NET, .NET Framework SDK, Authenticode, Products that contain the .NET CLR, Compact CLR or CLI.
A method of providing trusted intermediary services including the following steps:	
at a first apparatus, receiving an item from a second apparatus;	The item is an unsigned .NET assembly, which can include, but not be limited to, a Web control, multi-file assembly or component. Within the development environment, multiple assembly builders (working on a second apparatus) will send their unsigned assembly to a secure location (first apparatus) containing the entity's private signing key. An example entity would be a software publisher. .NET Security Framework – pg 130-1 Describes this exact practice and further explains the "Delay Signing Assemblies" feature of .NET that accommodates the fact that "many publishers will keep the private key in a secure location, possibly embedded in specially designed cryptographic hardware." "Delay signing is a technique used by developers whereby the public key is added to the assembly name as before, granting the assembly its unique identity, but no signature is computed. Thus, no private key access is necessary."
associating authentication information with said item;	Strong naming the assembly binds the entity's/publisher's name into the assembly. The public portion of the key used to strongly name the assembly is placed in the assembly manifest. Other assemblies or applications can contain references to the strong names of strongly named assemblies such as in the case of applications that contain references to a set of compliant .NET core libraries. Strong naming compliant .NET core libraries with the European Computers Manufactures Association's (ECMA) key is a way to allow any publisher to develop compliant .NET core libraries that can be authenticated by other applications.

	.NET Security Framework – pg 124 "Strong naming is a process whereby an assembly name can be further qualified by the identity of the publisher." .NET Security Framework – pg 133 The publisher must advertise its public key or keys in an out-of-band fashion (such as documentation shipped with the product or on the company Web site) .NET Security Framework – pg 130 The goal of the ECMA key is to allow a slightly more generalized strong name binding than usual, namely allowing binding to the publisher of the runtime in use, rather than to a fixed publisher.
incorporating said item into a secure digital container;	Signing the assembly places it in a secure container. .NET Framework Security – pg 527 Strong named assemblies cannot be modified in any manner without destroying the strong name signature. Applied Microsoft .NET Framework Programming – pg 89 *Strongly Named Assemblies Are Tamper-Resistant* When the assembly is installed into the GAC, the system hashes the contents of the file containing the manifest and compares the hash value with the RSA digital signature value embedded within the PE file (after unsigning it with the public key). If the values are identical, the file's contents haven't been tampered with and you know that you have the public key that corresponds to the publisher's private key. In addition, the system hashes the contents of the assembly's other files and compares the hash values with the hash values stored in the manifest file's FileDef table. If any of the hash values don't match, at least one of the assembly's files has been tampered with and the assembly will fail to install into the GAC.
associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;	A .NET assembly includes imperative and declarative statements/rules that will govern its access or use. For example, role-based security or strong name demands in the assembly can be the first rule. MSDN on Role-Based Security Applications that implement role-based security grant rights based on the role

	associated with a principal object. The principal object represents the security context under which code is running. The PrincipalPermission object represents the identity and role that a particular principal class must have to run. To implement the PrincipalPermission class imperatively, create a new instance of the class and initialize it with the name and role that you want users to have to access your code. MSDN on StrongNameIdentityPermission StrongNameIdentityPermission class defines the identity permission for strong names. StrongNameIdentityPermission uses this class to confirm that calling code is in a particular strong-named assembly.
transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;	The third apparatus is a user computer or an application server. The software publisher transmitting the .NET assembly to an end-user with a CLR. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.
said third apparatus receiving said secure digital container and said first rule;	The end-user receiving the signed assembly.
said third apparatus checking said authentication information; and	The .NET Framework, when the assembly is installed into the global assembly cash (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using the private key that corresponds to the public key of the publisher. Applied Microsoft .NET Framework Programming – pg 89 *Strongly Named Assemblies Are Tamper-Resistant* As above. .NET Framework Security – pg 128 The verification of any strong name assemblies is performed automatically when needed by the .NET Framework. Any assembly claiming a strong name but

	failing verification will fail to install into the global assembly or download cache or will fail to load at runtime.
said third apparatus performing at least one action on said item, said at least one action being governed, at least in part, by said first rule and by a second rule resident at said third apparatus prior to said receipt of said secure digital container and said first rule, said action governance occurring at least in part in said protected processing environment.	Within the CLR (protected processing environment), the execution of the program will depend upon whether the user is of the "role" required of the assembly or whether the calling assembly is from a strong-named assembly specified in the "item" assembly (alternate first rules) and only if assembly complies with the local code access security policy (second rule), as an example of one of the types of rules that .NET Framework allows to be resident on the third apparatus..

127. A method as in claim 126, in which said authentication information at least in part identifies said first apparatus and/or a user of said first apparatus.	The user of the first apparatus is the developer at the assembly developer. Strong naming binds the publisher's name to assembly.

LaMacchia, Brian, etc, .NET Framework Security, Addison-Wesley, 2002
Richter, Jeffrey, Applied Microsoft .NET Framework Programming, Microsoft Press, 2002

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,253,193

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
1	Infringing products include Windows Media Player and Windows Media Rights Manager SDK
A method comprising:	
(a) receiving a digital file including music;	Reference is made to the Windows Media Rights Manager SDK Programming Reference ("WMRM SDK"), attached hereto as Exhibit A. Media Player infringement analysis is set forth herein using the example of a music file downloaded and transferred to a portable audio player. Consumer receives a Windows Media file (WMRM SDK, Step 3)
(b) storing said digital file in a first secure memory of a first device;	Windows Media file is stored in consumer's computer and all use of it is securely managed by the Secure Content Manager in Windows Media Player.
(c) storing information associated with said digital file in a secure database stored on said first device, said information including at least one budget control and at least one copy control, said at least one budget control including a budget specifying the number of copies which can be made of said digital file; and said at least one copy control controlling the copies made of said digital file;	License is stored in the License Store (WMRM SDK, Step 5); license includes Rights which may include AllowTransfertoNonSDMI, AllowTransfertoSDMI, (or Allow Transfer to WM-D-DRM-Compliant devices or other types of devices), and TransferCount- the number of times a piece of content may be transferred to the device (a transfer budget).
(d) determining whether said digital file may be copied and stored on a second device based on at least said copy control;	Windows Media Rights Manager enforces the license restrictions
(e) if said copy control allows at least a portion of said digital file to be copied and stored on a second device,	Windows Media Rights Manager determines whether the AllowTransferToNonSDMI or AllowTransferToSDMI rights are present.(Or, Allow Transfer to WM-D-DRM-Compliant devices or other types of devices.)
(1)copying at least a portion of said digital file;	Transfer to the SDMI or non-SDMI portable device (Allow Transfer to WM-D-DRM-Compliant devices or other types of devices), if allowed by Windows Media Rights Manager
(2)transferring at least a portion of said digital file to a second device including a memory and an audio and/or video output;	Portable device necessarily includes at least a memory and audio output
(3)storing said digital file in said memory of said second device; and	Music file is transferred to the portable device
(4)including playing said music through said audio output.	Portable device plays the music
2. A method as in claim 1, further comprising:	
(a) at a time substantially contemporaneous with said transferring step, recording in said	Counter reflecting TransferCount is decremented by Windows Media Rights

first device information indicating that said transfer has occurred.	Manager
3. A method as in claim 2, in which:	
(a) said information indicating that said transfer has occurred includes an encumbrance on said budget.	Counter decrement reduces the allowable number of budgeted transfers
4. A method as in claim 3, in which:	
(a) said encumbrance operates to reduce the number of copies of said digital file authorized by said budget.	Counter decrement reduces the allowable number of budgeted transfers

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,253,193

	Infringing products include Windows Media Player and Windows Media Rights Manager SDK
11. A method comprising:	
(a) receiving a digital file;	Consumer receives a Windows Media file (WMRM SDK, Step 3)
(b) storing said digital file in a first secure memory of a first device;	Windows Media file is stored in consumer's computer and all use of it is securely managed by the Secure Content Manager in Windows Media Player.
(c) storing information associated with said digital file in a secure database stored on said first device, said information including a first control;	License information is stored in the License Store (WMRM SDK, Step 10), license information includes Rights. License Rights may include AllowTransferToNonSDMI, AllowTransferToSDMI (Allow Transfer to WM-D-DRM-Compliant devices or other types of devices), TransferCount
(d) determining whether said digital file may be copied and stored on a second device based on said first control,	WMRM determines whether transfer rights are included in license (WMRM SDK, Step 5)
(1) said determining step including identifying said second device and determining whether said first control allows transfer of said copied file to said second device, said determination based at least in part on the features present at the device to which said copied file is to be transferred;	Portable Device Service Provider Module identifies the portable device as either SDMI-compliant or non-SDMI-compliant (or WM-D-DRM Compliant or other types of supported devices) and provides this information to Windows Media Device Manager, which allows the transfer based on whether the device identification matches the License Right.
(e) if said first control allows at least a portion of said digital file to be copied and stored on a second device,	If Windows Media Rights Manager determines whether the AllowTransferToNonSDMI or AllowTransferToSDMI rights are present (or Allow Transfer to WM-D-DRM-Compliant devices or other types of devices), the following steps are performed:
(1) copying at least a portion of said digital file;	Transfer to the SDMI or non-SDMI (Allow Transfer to WM-D-DRM-Compliant or other) portable device, if allowed by Windows Media Rights Manager
(2) transferring at least a portion of said digital file to a second device including a memory and an audio and/or video output;	Portable device necessarily includes at least a memory and audio output
(3) storing said digital file in said memory of said second device; and	Music file is stored in the portable device
(4) rendering said digital file through said output.	Portable device plays the music

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,253,193

	Product infringing: Windows Media Player, Windows Media Player, Windows Media Rights Manager SDK
15. A method comprising:	
(a) receiving a digital file;	Consumer receives a Windows Media file ((WMRM SDK, Step 3)
(b) an authentication step comprising:	
(1) accessing at least one identifier associated with a first device or with a user of said first device; and	License includes identity of user's Windows Media Player. WM Players capable of playing protected content must be individualized. They contain a unique (Individualized) DRM client component to which protected WMA content licenses are bound. Content licenses are bound to this DRM individualization module as the result of a challenge sent from the Client to the WMLM service. The challenge contains information about Individualized DRM Client (in the form of an encrypted Client ID) and capabilities of the machine (e.g. support for Secure Audio Path (SAP), version of the WMRM SDK supported in the player.
(2) determining whether said identifier is associated with a device and/or user authorized to store said digital file;	Music file cannot be used unless identifier indicated in License matches user's Windows Media Player identifier (that is, the Individualized DRM Client to which the license is bound must be the same one supported by the device).
(c) storing said digital file in a first secure memory of said first device, but only if said device and/or user is so authorized, but not proceeding with said storing if said device and/or user is not authorized;	Music file will not be processed through Windows Media Player, including protected rendering buffers, unless the identifiers match. Protected WMA file can be stored on client even if unauthorized but it cannot be decrypted and enter into the secure boundary (first secure memory) of the player unless appropriately licensed.
(d) storing information associated with said digital file in a secure database stored on said first device, said information including at least one control;	License includes Rights and is stored in the License Store, Rights may include AllowTransferToNonSDMI, AllowTransferToSDMI, (or Allow Transfer To WM-D-DRM-CompliantDevice or other device) TransferCount
(e) determining whether said digital file may be copied and stored on a second device based on said at least one control;	Windows Media Rights Manager enforces the license restrictions
(f) if said at least one control allows at least a portion of said digital file to be copied and stored on a second device,	If appropriate rights are present, the following steps are performed:
(1) copying at least a portion of said	Transfer to the SDMI or non-SDMI (or WM-

digital file;	D-DRM Compliant or other) portable device, if allowed by Windows Media Rights Manager
(2) transferring at least a portion of said digital file to a second device including a memory and an audio and/or video output;	Portable device necessarily includes at least a memory and audio output
(3) storing said digital file in said memory of said second device; and	Music file is stored in the portable device
(4) rendering said digital file through said output.	Portable device plays the music
16. A method as in claim 15, in which:	
said digital file is received in an encrypted form; and further comprising: decrypting said digital file after said authentication step and before said step of storing said digital file in said memory of said first device.	Protected Windows Media File is encrypted. WMP will not decrypt file until license is processed. Licenses are bound to Individualization DLLs, which are bound to Hardware ID. Ind. DLL and Hardware ID must be verified as the Ids to which the license is bound – this is the authentication process. (Recall that this module was created based in part on receipt of the Client Hardware ID or fingerprint and the license was create based in part on receipt of a challenge from the client indicating the security properties (SAP-ready, SDK support, etc.) of the client).

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,253,193

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
19.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
A method comprising:	
receiving a digital file at a first device;	Receiving a digital file such as a Word Document, email, Excel spreadsheet, PowerPoint presentation, or other content at a recipient's device. Such content may be received via email, received on removable media, such as floppy disk, downloaded and viewable by Internet Explorer, *e.g.*, a web page possibly containing graphics and/or audio data, etc.
establishing communication between said first device and a clearinghouse located at a location remote from said first device;	If the digital file is subject to rights management, and the recipient tries to open the digital file in an IRM-enabled application, the IRM-enabled application contacts a remote RMS, *i.e.*, clearinghouse for a use license.
said first device obtaining authorization information including a key from said clearinghouse;	If the recipient is authorized to access or use the digital file, the RMS creates a license for the digital file. The RMS then seals a key inside the license so that only the recipient canaccess or use the digital file. Finally, the RMS sends the license back to the recipient.
said first device using said authorization information to gain access to or make at least one use of said first digital file, including using said key to decrypt at least a portion of said first digital file; and	The recipient's device then uses the key in the license to gain access or decrypt a portion of the digital file.
receiving a first control from said clearinghouse at said first device;	The license received from the RMS at the recipient's device contains at least one control, such as restricting the ability to print, forward, or edit.
storing said first digital file in a memory of said first device;	The digital file is stored in the memory of the said recipient's device, such as in RAM, on a hard drive, etc.
using said first control to determine whether said first digital file may be copied and stored on a second device;	The at least one control in the license limits copying the digital file. Such controls are set when the digital file was authored. For example, when the digital file is authored, the IRM-enabled application presented the author with a list of policy templates with different rights levels. The author selected an appropriate rights level which may for instance, allow other users in the system to open and read the document, but not

	to modify it, copy text from it, or forward it. These rights or controls are then associated with the digital file. When an attempt is made to access the digital file, the RMS determines the recipient's rights based on the recipient's identity and the policies or controls associated with the digital file.
if said first control allows at least a portion of said first digital file to be copied and stored on a second device,	If the control in the license allows copying the digital file to a second device, then at least a portion of the digital file is copied,
copying at least a portion of said first digital file;	such as by transferring or forwarding the digital file in an email message;
transferring at least a portion of said first digital file to a second device including a memory and an audio and/or video output;	A portion of the digital file is then transferred to a second device, such as a personal computer or portable device. The second device includes a memory and an audio and/or video output. The memory may be a hard-drive, RAM, CD, DVD, or other storage. The audio and/or video output may be speakers and/or a video monitor.
storing said first digital file portion in said memory of said second device; and	The digital file is stored in the second device's memory.
rendering said first digital file portion through said output.	The digital file is rendered through the output, such as played through the speakers and/or displayed on the video monitor. For example, a Word document is displayed on the screen of the video monitor.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,253,193

	Infringing products include Windows Media Player, Windows Media Rights Manager SDK
19. A method comprising:	
(a) receiving a digital file at a first device;	WMRM SDK, Step 3.
(b) establishing communication between said first device and a clearinghouse located at a location remote from said first device;	WMRM SDK, Step 6.
(c) said first device obtaining authorization information including a key from said clearinghouse;	WMRM SDK, Step 9. [License contains the key]
(d) said first device using said authorization information to gain access to or make at least one use of said first digital file, including using said key to decrypt at least a portion of said first digital file; and	WMRM SDK, Step 11.
(e) receiving a first control from said clearinghouse at said first device;	WMRM SDK, Steps 8-9.
(f) storing said first digital file in a memory of said first device;	WMRM SDK, Step 3.
(g) using said first control to determine whether said first digital file may be copied and stored on a second device;	At least the following WMRMRights Object properties meet this limitation: AllowTransferToNonSDMI, AllowTransferToSDMI (or AllowTransfer To WM-D-DRM-Compliant Device or other) and TransferCount
(h) if said first control allows at least a portion of said first digital file to be copied and stored on a second device,	This and all subsequent claim steps occur when the condition specified in the WMRMRights Object property is met
(i) copying at least a portion of said first digital file;	Transfer to the SDMI or non-SDMI (or WM-D-DRM Compliant) portable device, if allowed by Windows Media Rights Manager
(j) transferring at least a portion of said first digital file to a second device including a memory and an audio and/or video output;	Portable device necessarily includes at least a memory and audio output
(k) storing said first digital file portion in said memory of said second device; and	Music file is stored in the portable device
(l) rendering said first digital file portion through said output.	Portable device plays the music

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,253,193

	Infringing products include Windows Media Player, Windows Media Player, Windows Media Rights Manager SDK
51. A method comprising:	
(a) receiving a digital file at a first device;	WMRM SDK, Step 3.
(b) establishing communication between said first device and a clearinghouse located at a location remote from said first device;	WMRM SDK, Step 6.
(c) said first device obtaining authorization information from said clearinghouse; and	WMRM SDK, Step 9.
(d) said first device using said authorization information to gain access to or make at least one use of said first digital file;	WMRM SDK, Step 11.
(e) storing said first digital file in a memory of said first device;	WMA file stored on client
(f) using at least a first control to determine whether said first digital file may be copied and stored on a second device, said determination based at least in part on (1) identification information regarding said second device, and (2) the functional attributes of said second device;	If device is based on WM D-DRM, it has a certificate that is used to identify the device as compliant as well as the device's security level. The security level indicates support on the device for such attributes as an internal clock.
(g) if, based at least in part on said identification information, said first control allows at least a portion of said first digital file to be copied and stored on a second device,	If License specifies that transfer of protected WMA file to WM-D-DRM-Compliant device is allowed, transfer may occur.
(h) copying at least a portion of said first digital file;	If transfer is a licensed right as indicated in the license, the song is copied to the device via Windows Media Device Manager.
(i) transferring at least a portion of said first digital file to a second device including a memory and an audio and/or video output;	Windows Media Device Manager transfers the content to the device:
(j) storing said first digital file portion in said memory of said second device; and	WMA file is stored on device
(k) rendering said first digital file portion through said output.	WMA file is rendered.

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
33.	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.
A data processing arrangement comprising at least one storing arrangement that at least temporarily stores a first secure container comprising first protected data and a first set of rules governing use of said first protected data,	The first protected data is an ActiveX control. The first alternative for the first secure container is the signed .msi in which the ActiveX developer packaged the ActiveX control. The first set of rules is the conditional syntax statements of the signed .msi file. The second alternative for the first secure container is the signed and licensed ActiveX control. The first set of rules is the license support code in the ActiveX control. A third alternative for the first container is a signed cabinet file containing a (signed or unsigned) ActiveX control with license support code. The first set of rules is the license support code in the ActiveX control.
and at least temporarily stores a second secure container comprising second protected data different from said first protected data and a second set of rules governing use of said second protected data; and	The second protected data is the application developer's application that includes/uses the ActiveX control. The application developer's signed .msi file (second secure container) contains the application (second protected data). The second set of rules is the signed .msi file's conditional syntax statements that will be governed the offer/installation of the application.
a data transfer arrangement, coupled to at least one storing arrangement, for transferring at least a portion of said first protected data and a third set of rules governing use of said portion of said first protected data to said second secure container,	Placing the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container) that itself is included in the application's signed .msi file (second secure container). The third set of rules is the license support code in the ActiveX control.
further comprising	
means for creating and storing, in said at least one storing arrangement, a third secure container;	The ability of the application developer to package files in signed cabinet files.

said data transfer arrangement further comprising means for transferring said portion of said first protected data and said third set of rules to said third secure container, and means for incorporating said third secure container within said second secure container.	The third secure container is a cabinet file signed by the application developer and including at least the licensed ActiveX control (first protected information. The licensing support code in the ActiveX control when its developer added licensing support to the ActiveX control is the third set of rules.
34. A data processing arrangement as in claim 33 further comprising means for applying said third set of rules to govern at least one aspect of use of said portion of said first protected data.	Before an ActiveX control will create a copy of itself, the calling application has to pass a license key to the ActiveX control. The license support code in the ActiveX control (third rule set) evaluates the authenticity of the calling application's request.
35. A data processing arrangement as in claim 34 further comprising means for applying said second set of rules to govern at least one aspect of use of said portion of said first protected data.	Windows Installer operating system service enforces the conditional syntax statements of the application's signed .msi file. These statements govern the offer/installation of the ActiveX control.

41	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.
A method comprising performing the following steps within a virtual distribution environment comprising one or more electronic appliances and a first secure container, said first secure container comprising (a) a first control set, and (b) a second secure container comprising a second control set and first protected information:	The signed .msi file created by the ActiveX control developer is the first secure container. The conditional syntax statement(s) of the ActiveX control developer's signed .msi file is/are the first control set. The first protected information is the ActiveX control. The first alternative for the second secure container is the signed and licensed ActiveX control. The second control set is the license support code in the ActiveX control. The second alternative for the second secure container is a signed cabinet file containing the (signed or unsigned) ActiveX control. The second control set is the license support code in the ActiveX control.
using at least one control from said first control set or said second control set to govern at least one aspect of use of said first protected information while said first protected information is contained within said first secure container;	The ActiveX control developer's conditional syntax statements (first control set) in the ActiveX developer's signed .msi file govern the offer/installation of the ActiveX control while it is in its signed .msi file. Alternately, the license support code (second control set) in the ActiveX control governs use of the licensed ActiveX control.
creating a third secure container comprising a third control set for governing at least one aspect of use of protected information contained within said third secure container;	The third secure container is a signed .msi file. The application developer packages its application in a signed .msi file (third secure container) and includes conditional syntax statements (third control set) in the signed .msi
incorporating a first portion of said first protected information in said third secure container, said first portion made up of some or all of said first protected information; and	Placing the ActiveX control into the application developer's signed .msi file (third secure container).
using at least one control to govern at least	The application developer's conditional

one aspect of use of said first portion of said first protected information while said first portion is contained within said third secure container.	syntax statement(s) in its signed .msi file govern the offer/installation ActiveX control while it is in the signed .msi file (third secure container).
42. A method as in claim 41, in which said first secure container further includes a fourth secure container comprising a fourth control set and second protected information and further comprising the following step:	The second protected information is a second ActiveX control. The first alternative for the fourth secure container is the signed and licensed second ActiveX control. The fourth control set is the license support code in the ActiveX control. The second alternative for the fourth secure container is a signed cabinet file containing the (signed or unsigned) second ActiveX control. The fourth control set is the license support code in the ActiveX control.
using at least one control from said first control set or said fourth control set to govern at least one aspect of use of said second protected information while said second protected information is contained within said first secure container.	The ActiveX control developer's conditional syntax statements (first control set) in the ActiveX developer's signed .msi file govern the offer/installation of the second ActiveX control while it is in its signed .msi file. Alternately, the license support code (second control set) in the ActiveX control governs use of the licensed ActiveX control.
47. A method as in claim 41, in which said step of creating a third secure container includes:	
creating said third control set by incorporating at least one control not found in said first control set or said second control set.	The application developer's conditional syntax statements are not found in either the first control set or the second control set.
52. A method as in claim 41 in which said step of creating a third secure container occurs at a first site, and further comprising:	
copying or transferring said third secure container from said first site to a second site located remotely from said first site.	The application developer at first site distributes its application to other sites.
53. A method as in claim 52 in which said first site is associated with a content distributor.	The application developer at the first site is the content distributor.
54. A method as in claim 53 in which said second site is associated with a user of	The application developer distributes the application to end-users.

content.	
55. A method as in claim 54 further comprising the following step:	
said user directly or indirectly initiating communication with said first site.	For Internet downloads, the user initiates the communication with the first site.
64. A method as in claim 54 in which said third control set includes one or more controls at least in part governing the use by said user of at least a portion of said first portion of said first protected information.	The application developer's conditional syntax statements (third control set) govern the installation of the ActiveX control (first protected information).
76. A method as in claim 41 in which said creation of said third secure container further comprises using a template which specifies one or more of the controls contained in said third control set.	The third secure container is the application developer's signed .msi file and the third control set is the conditional syntax statements in that file. Microsoft supplies several template .msi databases for use in authoring installation packages. The UISample.msi is the template recommended in the "An Installation Example" on MSDN. This template msi files contains several default conditional syntax statements. At least two of these conditional syntax statements directly govern the installation by blocking progress until the EULA is accepted.
78. A method as in claim 52 in which said creation of said third secure container further comprises using a template which specifies one or more of the controls contained in said third control set.	The third secure container is the application developer's signed .msi file and the third control set is the conditional syntax statements in that file. Microsoft supplies several template .msi databases for use in authoring installation packages. The UISample.msi is the template recommended in the "An Installation Example" on MSDN. This template msi files contains several default conditional syntax statements. At least two of these conditional syntax statements directly govern the installation by blocking progress until the EULA is accepted.

81.	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.
A data processing arrangement comprising:	
a first secure container comprising first protected information and a first rule set governing use of said first protected information;	The first alternative for the first secure container is the ActiveX control developer's signed .msi file containing a licensed ActiveX control (the first protected information). The conditional syntax statements of the signed .msi file are the first rule set. The second alternative for the first secure container is the signed cabinet file containing the ActiveX control. The license support code in the ActiveX control is the first rule set. The third alternative for the first secure container is the licensed and signed ActiveX control governed by license support code in the ActiveX control.
a second secure container comprising a second rule set;	The second secure container is the signed .msi file which the application developer package its application. The second rule set is the conditional syntax statements of the application developer's signed .msi file.
means for creating and storing a third secure container; and	The third container is a signed cabinet file containing at least the ActiveX control.
means for copying or transferring at least a portion of said first protected information and a third rule set governing use of said portion of said first protected information to said second secure container, said means for copying or transferring comprising:	Putting the licensed ActiveX control (first protected information) in a signed cabinet file (third secure container). The licensing support code in the ActiveX control is third rule set.
means for incorporating said third secure container within said second secure container.	Packaging the signed cabinet file in the signed .msi file.

82. A data processing arrangement as in claim 81 further comprising:	
means for applying at least one rule from said third rule set to at least in part govern at least one factor related to use of said portion of said first protected information.	The third rule set ensures the user is licensed.

83. A data processing arrangement as in claim 82 further comprising:	

means for applying at least one rule from said second rule set to at least in part govern at least one factor related to use of said portion of said first protected information.	The second rule set governs the offer/installation of first protected information.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.

INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,915,019

85.	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.
A method comprising the following steps:	
creating a first secure container comprising a first rule set and first protected information;	The first protected information is the ActiveX control. The first alternative for the first secure container is the signed and licensed ActiveX control. The first rule set is the license support code in the ActiveX control. The second alternative for the first secure container is an (signed or unsigned) ActiveX control with license support contained within a signed cabinet file. The first rule set is the ActiveX license support code.
storing said first secure container in a first memory;	The first secure container is stored at the ActiveX control developer's location.
creating a second secure container comprising a second rule set;	The second secure container is the application developer's signed .msi file. The conditional syntax statements of the signed .msi file are the second rule set.
storing said second secure container in a second memory;	The second secure container is stored at the application developer's location.
copying or transferring at least a first portion of said first protected information to said second secure container, said copying or transferring step comprising:	The ActiveX control developer packages the control in a signed .msi file for distribution to the application developer's site.
creating a third secure container comprising a third rule set;	The third secure container is the ActiveX control developer's signed .msi file containing a licensed ActiveX control. The conditional syntax statements of the signed .msi file are the third rule set.
copying said first portion of said first protected information;	In preparation for using a msi authoring tool, such as Microsoft's Orca, copying the ActiveX control to a package staging area.
transferring said copied first portion of said first protected information to said third secure container; and	Using msi authoring tool to import the control into the signed .msi file.
copying or transferring said copied first portion of said first protected information from said third secure container to said second secure container.	The application developer installs the ActiveX control, which involves removing it from the ActiveX developer's signed .msi file and installing it into its environment. Subsequently, the

	application developer places the ActiveX control into its signed .msi file when it is packaging its application.

87. A method as in claim 85 in which said copied first portion of said first protected information consists of the entirety of said first protected information.	The entire ActiveX control is copied.

89. A method as in claim 85 in which	
said first memory is located at a first site,	The first memory is located at the ActiveX control developer's site.
said second memory is located at a second site remote from said first site, and	The second memory is located at the application developer's site.
said step of copying or transferring said first portion of said first protected information to said second secure container further comprises copying or transferring said third secure container from said first site to said second site.	The ActiveX control developer's signed .msi file is transferred from its site to the site of the application developer.

85. (alternate infringing scenario)	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.
A method comprising the following steps:	
creating a first secure container comprising a first rule set and first protected information;	The first protected information is the ActiveX control. The first alternative for the first secure container is the signed and licensed ActiveX control. The first rule set is the license support code in the ActiveX control. The second alternative for the first secure container is a (signed or unsigned) ActiveX control with license support contained within a signed cabinet file. The first rule set would remain the ActiveX license support code. The third alternative for the first secure container is a signed msi file in which the ActiveX control developer packaged its ActiveX control. The first rule set is the conditional syntax statement(s) of the signed msi file.
storing said first secure container in a first memory;	The first secure container is stored at the ActiveX control developer's location.
creating a second secure container comprising a second rule set;	The second secure container is the application developer's signed .msi file. The conditional syntax statements of the signed .msi file are the second rule set.
storing said second secure container in a second memory;	The second secure container is stored at the application developer's location.
copying or transferring at least a first portion of said first protected information to said second secure container, said copying or transferring step comprising:	The ActiveX control is placed in a cabinet file signed by the application developer and the signed cabinet file is placed in a .msi file signed by the application developer.
creating a third secure container comprising a third rule set;	The third secure container is signed cabinet file in which the application developer placed licensed ActiveX. The third rule set is the license support code in the ActiveX control.
copying said first portion of said first protected information;	Copying ActiveX control.
transferring said copied first portion of said first protected information to	Transferring ActiveX control to signed cabinet file.

said third secure container; and	
copying or transferring said copied first portion of said first protected information from said third secure container to said second secure container.	The application developer places the signed cabinet file into its signed .msi file when it is packaging its application.

87. A method as in claim 85 in which said copied first portion of said first protected information consists of the entirety of said first protected information.	The entire ActiveX control is copied.

93. A method as in claim 85 in which said step of copying transferring said copied first portion of said first protected information from said third secure container to said second secure container further comprises storing said third secure container in said second secure container.	The ActiveX control is placed in a cabinet file signed by the application developer and the signed cabinet file is placed in a .msi file signed by the application developer.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.

INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,915,019

1:	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer technology.
A method of operating on a first secure container arrangement having a first set of controls associated therewith, said first secure container arrangement at least in part comprising a first protected content file, said method comprising the following steps performed within a virtual distribution environment including at least one electronic appliance:	The first protected content is a signed and licensed .NET component used by the .NET assembly. The .NET assembly is distributed with a signed and governed .msi file. The second protected content is another signed and licensed .NET component that is used by the .NET assembly.
using at least one control associated with said first secure container arrangement for governing, at least in part, at least one aspect of use of said first protected content file while said first protected content file is contained in said first secure container arrangement;	The first protected content is signed and licensed .NET component (first secure container) contained within the .NET assembly. The one control is a declarative statement(s) within the assembly's header.
creating a second secure container arrangement having a second set of controls associated therewith, said second set of controls governing, at least in part, at least one aspect of use of any protected content file contained within said second secure container arrangement;	The protected content is the same as the first protected content plus the additional implementation information included in the signed .msi file. The second secure container is the signed .msi file created for the .NET assembly. The signed .msi file's conditional syntax statements are the second set of controls that control the offer/installation of the .NET assembly.
transferring at least a portion of said first protected content file to said second secure container arrangement, said portion made up of at least some of said first protected content file; and	The entire .NET assembly is included in the signed .msi file. Packaging the .NET assembly in the signed .msi file involves the following process steps. In preparation for using a msi authoring tool, such as Microsoft's Orca, copying the .NET component to a package staging area. Using msi authoring tool to import the .NET component into the signed .msi file.
using at least one rule to govern at least one aspect of use of said first protected content file portion while said portion is contained within said second secure container arrangement:	The conditional syntax statement(s) of the signed .msi file (second secure container) control(s) the offer/installation of the .NET assembly.
in which	
said first secure container arrangement comprises a third secure container	The first alternative for the third secure container is a licensed and signed .NET

arrangement comprising a third set of controls and said first protected content file, and	component governed by the set of declarative statements comprising the LicenseProviderAttribute (third set of controls). The second alternative for the third secure container is a .NET component whose hash is included in the header of the .NET assembly. The set of declarative statements comprising the LicenseProviderAttribute is the third set of controls.
said first secure container arrangement further comprises a fourth secure container arrangement comprising a fourth set of controls and a second protected content file.	The first alternative for the fourth secure container is another licensed and signed .NET component governed by the set of declarative statements comprising the LicenseProviderAttribute (fourth set of controls). The second alternative for the fourth secure container is the container created when the hash of the .NET component is included in the header information of the .NET assembly. The set of declarative statements comprising the LicenseProviderAttribute is the fourth set of controls.

33.	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer technology.
A data processing arrangement comprising at least one storing arrangement that at least temporarily stores a first secure container comprising first protected data and a first set of rules governing use of said first protected data,	The first protected information is the .NET component. The first alternate for the first secure container is the signed .msi file in which the .NET component developer packaged its .NET component. The first set of rules is the conditional syntax statements of the signed .msi file. The second alternative for the first secure container is a licensed and signed .NET component governed by the set of declarative statements comprising the LicenseProviderAttribute of the .NET component (first set of controls). The third alternative for the first container is a signed cabinet file containing a (signed or unsigned) .NET component with license support. The first set of controls is the set of declarative statements comprising the LicenseProviderAttribute of the .NET component.
and at least temporarily stores a second secure container comprising second protected data different from said first protected data and a second set of rules governing use of said second protected data; and	The second protected data is the .NET assembly developer's assembly that includes/uses the .NET component. The first alternative for the second secure container is a signed .msi file in which the .NET assembly developer packaged its multi-file assembly (second protected data). The second set of rules is the conditional syntax statements of the signed .msi file that governs the offer/installation of the .NET assembly. The second alternative for the second secure container is a signed .NET assembly. The second set of rules is the declarative rules within the assembly's header.
a data transfer arrangement, coupled to at least one storing arrangement, for	The third secure container is a signed .NET assembly governed by declarative rules in

transferring at least a portion of said first protected data and a third set of rules governing use of said portion of said first protected data to said second secure container,	its header (third set of rules). An alternative third rule set is the set of declarative statements comprising the LicenseProviderAttribute. The .NET assembly includes the .NET component. The secure .NET assembly is included in a signed .msi file (second secure container). An alternative third secure container is the container created by hashing the .NET component and including the hash in the header information of a .NET assembly. The .NET component is included in the signed and governed .NET assembly (second secure container). The third set of rules is the set of declarative statements comprising the LicenseProviderAttribute. An alternative third secure container is a signed cabinet file containing the .NET component and which is destined for a signed .msi file (second secure container). The third set of rules is the set of declarative statements comprising the LicenseProviderAttribute.
further comprising	
means for creating and storing, in said at least one storing arrangement, a third secure container;	The first alternative for the third secure container is a signed .NET assembly. In this case, the second secure container is the signed .msi file. The second alternative for the third container is the container created by including a hash of the .NET component in the header information of a .NET assembly. In this case, the second secure container is either the signed .msi file or the signed .NET assembly. The third alternative for the third container is a cabinet file signed by the .NET assembly developer containing the .NET assembly and/or the .NET component. In this case the signed .msi file is the second secure container.
said data transfer arrangement further comprising means for transferring said portion of said first protected data and said third set of rules to said third secure container, and means for incorporating said third secure container within said second secure container.	The first alternative for the third secure container is the signed .NET assembly, which includes and/or uses the licensed .NET component (first protected information). The third set of rules is a declarative rule within the .NET assembly's header. The .NET assembly is placed in a signed .msi file (second secure container).

	The second alternative for the third secure container is the container that results when the hash of the .NET component is added to the .NET assembly header information. The third set of rules is the set of declarative statements comprising the LicenseProviderAttribute added to the assembly. The third alternative for the third secure container is a cabinet file signed by the .NET assembly developer containing the .NET assembly and/or the .NET component. The third set of rules is a declarative rule(s) within the .NET assembly's header and/or the set of declarative statements comprising the LicenseProviderAttribute added to the assembly
34. A data processing arrangement as in claim 33 further comprising means for applying said third set of rules to govern at least one aspect of use of said portion of said first protected data.	When the third rule set is the declarative statement(s) of the assembly header, the runtime CLR enforces the statements. When the third set of rules is the set of declarative statements comprising the LicenseProviderAttribute added to the assembly, the license support code in the .NET component evaluates the authenticity of the calling assembly's request.
35. A data processing arrangement as in claim 34 further comprising means for applying said second set of rules to govern at least one aspect of use of said portion of said first protected data.	When the second set of rules is the conditional syntax statements of the signed .msi file, the Windows Installer operating system service enforces the conditional syntax statements of .NET assembly's signed .msi file, which govern the offer/installation of the .NET component. When the second set of rules is the declarative statement(s) within the assembly's header, the runtime CLR enforces the statements.

41.	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer technology.
A method comprising performing the following steps within a virtual distribution environment comprising one or more electronic appliances and a first secure container, said first secure container comprising (a) a first control set, and (b) a second secure container comprising a second control set and first protected information:	The signed .msi file created by the .NET component developer is the first secure container. The first conditional syntax statement(s) of the .NET component developer's signed .msi file is/are the first control set. The first protected information is the .NET component. The first alternative for the second secure container is the signed and licensed .NET component. The second control set is the set of declarative statements comprising the LicenseProviderAttribute. The second alternative for the second secure container is a signed cabinet file. The second control set remains the set of declarative statements comprising the LicenseProviderAttribute.
using at least one control from said first control set or said second control set to govern at least one aspect of use of said first protected information while said first protected information is contained within said first secure container;	The .NET component developer's conditional syntax statements (first control set) in its signed .msi file governs the offer/installation of the .NET component while it is in the signed .msi file. Alternately, the set of declarative statements comprising the LicenseProviderAttribute (second control set) of the licensed .NET component governs use of the .NET component.
creating a third secure container comprising a third control set for governing at least one aspect of use of protected information contained within said third secure container;	The first alternative for the third secure container is a signed .NET assembly, the protected information is the .NET component and the third control set is the declarative statement(s) within the .NET assembly's header. The second alternative for the third secure container is a signed .msi file in which the .NET assembly developer packages its .NET assembly and the third control set is the conditional syntax statement(s) in the signed .msi file.

incorporating a first portion of said first protected information in said third secure container, said first portion made up of some or all of said first protected information; and	In the first alternative, placing the .NET component into the signed .NET assembly. In the second alternative, placing the .NET component into the. Net assembly developer's signed msi file.
using at least one control to govern at least one aspect of use of said first portion of said first protected information while said first portion is contained within said third secure container.	In the first alternative, the .NET assembly developer's declarative statement(s) within the .NET assembly's header govern(s) the use of the .NET component while it is in the signed .NET assembly. In the second alternative, the conditional syntax statements of the .NET assembly developer's signed .msi file govern the offer/installation of the .NET component while it is in the signed .msi file.

42. A method as in claim 41, in which said first secure container further includes a fourth secure container comprising a fourth control set and second protected information and further comprising the following step:	The second protected information is a second .NET component. The first alternative for the fourth secure container is the signed and licensed second .NET component. The fourth control set is the set of declarative statements comprising the LicenseProviderAttribute of the second .NET component. The second alternative for the fourth secure container is a second signed cabinet file. The fourth control set is the set of declarative statements comprising the LicenseProviderAttribute.
using at least one control from said first control set or said fourth control set to govern at least one aspect of use of said second protected information while said second protected information is contained within said first secure container.	The .NET component developer's conditional syntax statements (first control set) in its signed .msi file governs the offer/installation of the second .NET component while it is in the signed .msi file. Alternately, the set of declarative statements comprising the LicenseProviderAttribute (fourth control set) of the licensed second .NET component governs use of the second .NET component.

47. A method as in claim 41, in which said step of creating a third secure container includes:	
creating said third control set by incorporating at least one control not found in said first control set or said second control set.	The .NET assembly developer's declarative statements (first alternative for third control set) and/or the developer's conditional syntax statements (second alternative for the third control set) are not found in either

	the first control set or the second control set.
52. A method as in claim 41 in which said step of creating a third secure container occurs at a first site, and further comprising:	
copying or transferring said third secure container from said first site to a second site located remotely from said first site.	The .NET assembly developer at first site distributes its assembly to other sites.
53. A method as in claim 52 in which said first site is associated with a content distributor.	The .NET assembly developer's business module is used to create and distribute its assembly.
54. A method as in claim 53 in which said second site is associated with a user of content.	The .NET assembly developer distributes the assembly to end-users.
55. A method as in claim 54 further comprising the following step:	
said user directly or indirectly initiating communication with said first site.	For Internet downloads, the user initiates the communication with the first site.
64. A method as in claim 54 in which said third control set includes one or more controls at least in part governing the use by said user of at least a portion of said first portion of said first protected information.	When the third control set is the .NET assembly developer's declarative statement(s) within the .NET assembly's header, it governs the user's use of the .NET component (first protected information). When the third control set is the .NET assembly developer's conditional syntax statements of the .NET assembly developer's signed .msi file, it governs the user's offer acceptance/installation of the .NET component (first protected information).
76. A method as in claim 41 in which said creation of said third secure container further comprises using a template which specifies one or more of the controls contained in said third control set.	When the third secure container is the .NET assembly developer's signed .msi file and the third control set is the conditional syntax statements in that file. Microsoft supplies several template .msi databases for use in authoring installation packages. The UISample.msi is the template recommended in the "An Installation Example" on MSDN. This template msi files contains several default conditional syntax statements. At least two of these conditional syntax statements directly govern the installation by blocking progress until the EULA is accepted.

78. A method as in claim 52 in which said creation of said third secure container further comprises using a template which specifies one or more of the controls contained in said third control set.	When the third secure container is the .NET assembly developer's signed .msi file and the third control set is the conditional syntax statements in that file. Microsoft supplies several template .msi databases for use in authoring installation packages. The UISample.msi is the template recommended in the "An Installation Example" on MSDN. This template msi files contains several default conditional syntax statements. At least two of these conditional syntax statements directly govern the installation by blocking progress until the EULA is accepted.

81.	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer technology.
A data processing arrangement comprising:	
a first secure container comprising first protected information and a first rule set governing use of said first protected information;	The first protected information is the .NET component. The first alternative for the first secure container is the signed .msi file in which the .NET component developer packaged its assembly. The first rule set is the conditional syntax statements written by the .NET component developer and placed into the signed .msi file. The second alternative for the first secure container is the signed cabinet file containing the (signed or unsigned) .NET component. The set of declarative statements comprising the LicenseProviderAttribute when its developer added licensing support to the assembly is the first rule set. The third alternative for the first secure container is the licensed and signed .NET component governed by the set of declarative statements comprising the LicenseProviderAttribute (first rule set) added by the .NET component developer.
a second secure container comprising a second rule set;	The first alternative for the second secure container is the signed .msi file in which the .NET assembly developer packaged its .NET assembly. The second rule set is the conditional syntax statements written by the .NET assembly developer and placed into the signed .msi file. The second alternative for the second secure container is the signed .NET assembly. The second rule set is the declarative statements in the .NET assembly's header.
means for creating and storing a third secure container; and	When the second secure container is the signed msi file, the third secure container is the signed .NET assembly. When the second secure container is the

	signed .NET assembly, the third secure container a .NET component secured by placing it in a signed cabinet file or by including its hash in the header of the assembly.
means for copying or transferring at least a portion of said first protected information and a third rule set governing use of said portion of said first protected information to said second secure container, said means for copying or transferring comprising:	When the second secure container is the signed msi file and the third secure container is the signed .NET assembly, the third rule set is the set of declarative statements within the assembly's header. When the second secure container is the signed .NET assembly, the third rule set is the set of declarative statements comprising the LicenseProviderAttribute (third rule set) added to the .NET component by its developer.
means for incorporating said third secure container within said second secure container.	When the second secure container is the signed msi file and the third secure container is the signed .NET assembly, the assembly is placed in the signed .msi file. When the second secure container is the signed .NET assembly and the third secure container is a .NET component contained in a signed cabinet file or a .NET component whose hash is included in the header of the assembly, the third secure container is incorporated within the .NET assembly.

82. A data processing arrangement as in claim 81 further comprising:	
means for applying at least one rule from said third rule set to at least in part govern at least one factor related to use of said portion of said first protected information.	When the third rule set is declarative statements within the assembly's header, it governs the use of the .NET assembly which includes the first protected information. When the third rule set is the set of declarative statements comprising the LicenseProviderAttribute added by the .NET component by its developer, it ensures the user is licensed.

83. A data processing arrangement as in claim 82 further comprising:	
means for applying at least one rule from said second rule set to at least in part govern at least one factor related to use of said portion of said first protected information.	When the second rule set is the conditional syntax statements written by the .NET assembly developer and placed into the signed .msi file, it governs the offer/installation of the .NET component. When the second rule set is the declarative statements in the .NET assembly's header,

	it governs the use of the .NET assembly, which includes the first protected information.

85. A method comprising the following steps:	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer technology.
creating a first secure container comprising a first rule set and first protected information;	The first protected information is the .NET component. The first secure container is a signed .NET component (first protected information) governed by the set of declarative statements comprising the LicenseProviderAttribute (first rule set). The second alternative for the first secure container is a cabinet file signed by the .NET component developer containing a (signed or unsigned) .NET component with license support. The first rule set is the set of declarative statements comprising the LicenseProviderAttribute.
storing said first secure container in a first memory;	The first secure container is stored at the .NET component developer's location.
creating a second secure container comprising a second rule set;	The first alternative for the second secure container is a signed .NET assembly and the second rule set is declarative statement(s) within the assembly's header. The second alternative for the second secure container is the signed .msi file in which the .NET assembly developer packages its (signed or unsigned) assembly. The second rule set is the conditional syntax statement(s) written by the .NET assembly developer and placed into the signed .msi file.
storing said second secure container in a second memory;	The second secure container is stored at the .NET assembly developer's location.
copying or transferring at least a first portion of said first protected information to said second secure container, said copying or transferring step comprising:	The .NET component developer packages its module in a signed .msi file for distribution to the .NET assembly developer's site.
creating a third secure container comprising a third rule set;	The third secure container is the signed .msi file in which the .NET component developer packaged its .NET component. The third control set is the conditional syntax statements written by the .NET component developer and placed into the signed .msi file.
copying said first portion of said	In preparation for using a msi authoring

first protected information;	tool, such as Microsoft's Orca, copying the .NET component to a package staging area.
transferring said copied first portion of said first protected information to said third secure container; and	Using the msi authoring tool to import the .NET component into the signed .msi file.
copying or transferring said copied first portion of said first protected information from said third secure container to said second secure container.	The .NET assembly developer installs the .NET component, which involves removing it from the .NET component developer's signed .msi file and installing it into its environment. Subsequently, the .NET assembly developer places the .NET component into its .NET assembly and/or signed .msi file when it is packaging its .NET assembly.

87. A method as in claim 85 in which said copied first portion of said first protected information consists of the entirety of said first protected information.	The entire .NET component is copied.

89. A method as in claim 85 in which	
said first memory is located at a first site,	The first memory is located at the .NET component developer's site.
said second memory is located at a second site remote from said first site, and	The second memory is located at the .NET assembly developer's site.
said step of copying or transferring said first portion of said first protected information to said second secure container further comprises copying or transferring said third secure container from said first site to said second site.	The .NET component developer's signed .msi file is transferred from its site to the site of the .NET assembly developer.

94. A method as in claim 85 further comprising:	
creating a fourth rule set.	When the second secure container is not a signed .NET assembly, the fourth rule set is declarative statements within the assembly's header. When the second secure container is not the signed .msi file in which the .NET assembly developer packages its (signed or unsigned) assembly, the fourth rule set is the conditional syntax statements written by the .NET assembly developer and placed into the signed .msi file.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.

INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,915,019

85 (alternate infringing scenario)	
A method comprising the following steps:	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer technology.
creating a first secure container comprising a first rule set and first protected information;	The first protected information is the .NET component. The first alternative for the first secure container is the signed and licensed .NET component. The first rule set is the set of declarative statements comprising the LicenseProviderAttribute in the .NET component. The second alternative for the first secure container is a (signed or unsigned) .NET component with license support contained within a cabinet file signed by the .NET component developer. The first rule set is the set of declarative statements comprising the LicenseProviderAttribute in the .NET component. The third alternative for the first secure container is the signed .msi file in which the .NET component developer packaged its assembly. The first rule set is the conditional syntax statements written by the .NET component developer and placed into the signed .msi file.
storing said first secure container in a first memory;	The first secure container is stored at the .NET component developer's location.
creating a second secure container comprising a second rule set;	The first alternative for the second secure container is a signed .NET assembly and the second rule set is declarative statement(s) within the assembly's header. The second alternative for the second secure container is the signed .msi file in which the .NET assembly developer packages its (signed or unsigned) assembly. The second rule set is the conditional syntax statement(s) written by the .NET assembly developer and placed into the signed .msi file.
storing said second secure container in a second memory;	The second secure container is stored at the .NET assembly developer's location.
copying or transferring at least a first	The .NET assembly developer places the

portion of said first protected information to said second secure container, said copying or transferring step comprising:	.NET component into the third secure container, which is either a signed cabinet file or a signed .NET assembly.
creating a third secure container comprising a third rule set;	When the second secure container is the signed .msi file, the third secure container is the signed .NET assembly. The third rule set is the declarative statement(s) in the .NET assembly's header. When the second secure container is either a .NET assembly or the signed .msi file, the third secure container is a signed cabinet file in which the .NET assembly developer placed licensed .NET component. The third rule set is the set of declarative statements comprising the LicenseProviderAttribute in the .NET component.
copying said first portion of said first protected information;	Copying the .NET component to either the .NET assembly or to the signed cabinet file.
transferring said copied first portion of said first protected information to said third secure container; and	Transferring the .NET component to either the .NET assembly or the signed cabinet file.
copying or transferring said copied first portion of said first protected information from said third secure container to said second secure container.	When the second secure container is the signed .msi file and the third secure container is the signed .NET assembly, the .NET assembly is placed into the signed .msi file. When the second secure container is either the .NET assembly or the signed .msi file and the third secure container is the signed cabinet file, the signed cabinet file is placed into either the .NET assembly or the signed .msi file.

87. A method as in claim 85 in which said copied first portion of said first protected information consists of the entirety of said first protected information.	The entire .NET component is copied.

93. A method as in claim 85 in which	
said step of copying transferring said copied first portion of said first protected information from said third secure container to said second secure container further comprises storing said third secure container in said second secure container.	When the third secure container is the signed .NET assembly, it is placed in the signed .msi file. When the third secure container is a signed cabinet file, it can be placed in either the .NET assembly and/or the signed .msi file.

94. A method as in claim 85 further comprising:	
creating a fourth rule set.	When the second rule set is declarative statement(s) within the assembly's header.

	the fourth rule set is the conditional syntax statement(s) written by the .NET assembly developer and placed into the signed .msi file. When the second rule set is the conditional syntax statement(s) written by the .NET assembly developer and placed into the signed .msi file, the fourth rule set is declarative statement(s) within the assembly's header or the set of declarative statements comprising the LicenseProviderAttribute in the .NET component.

95. A method as in claim 94 further comprising:	
using said fourth rule set to govern at least one aspect of use of said copied first portion of said first protected information.	If the fourth rule set is the .NET assembly developer's declarative statement(s) within the .NET assembly's header, it governs the use of the .NET component. If the fourth rule set is the conditional syntax statements of the .NET assembly developer's signed .msi file, it governs the offer/installation of the .NET component.

85 (second alternate scenario for .NET)	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer technology.
A method comprising the following steps:	
creating a first secure container comprising a first rule set and first protected information;	The first protected information is a .NET component. The first alternative for the first secure container is the signed and licensed .NET component. The first rule set is the set of declarative statements comprising the LicenseProviderAttribute in the .NET component. The second alternative for the first secure container is a (signed or unsigned) .NET component with license support contained within a cabinet file signed by the .NET assembly developer. The first rule set is the set of declarative statements comprising the LicenseProviderAttribute in the .NET component. The third alternative for the first secure container is a .NET component whose hash is included in the assembly header of a .NET assembly. The first rule set is the set of declarative statements comprising the LicenseProviderAttribute in the .NET component.
storing said first secure container in a first memory;	The first secure container is stored at the .NET assembly developer's location.
creating a second secure container comprising a second rule set;	The second secure container is the signed .msi file in which the .NET assembly developer packages its signed assembly. The second rule set is the conditional syntax statement(s) written by the .NET assembly developer and placed into the signed .msi file.
storing said second secure container in a second memory;	The second secure container is stored at the .NET assembly developer's location.
copying or transferring at least a first portion of said first protected information to said second secure container, said copying or transferring step comprising:	The .NET assembly developer places the .NET component into the third secure container, which is the signed .NET assembly.
creating a third secure container comprising a third rule set;	The third secure container is a signed .NET assembly and the third rule set is

	declarative statement(s) within the assembly's header.
copying said first portion of said first protected information;	Copying the .NET component to the .NET assembly.
transferring said copied first portion of said first protected information to said third secure container; and	Transferring the .NET component to the .NET assembly.
copying or transferring said copied first portion of said first protected information from said third secure container to said second secure container.	When the second secure container is the signed .msi file and the third secure container is the signed .NET assembly, the .NET assembly is placed into the signed .msi file.

87. A method as in claim 85 in which said copied first portion of said first protected information consists of the entirety of said first protected information.	The entire .NET component is copied.

90. A method as in claim 85 in which	
said first memory and said second memory are located at the same site.	First and second memory is at the .NET assembly developer's location.

93. A method as in claim 85 in which	
said step of copying transferring said copied first portion of said first protected information from said third secure container to said second secure container further comprises storing said third secure container in said second secure container.	When the third secure container is the signed .NET assembly, it is placed in the signed .msi file.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 5,915,019

96. A method comprising performing the following steps within a virtual distribution environment comprising one or more electronic appliances and a first secure container, said first secure container comprising a first control set and first protected information:	A signed and licensed .NET component (first container) is part of a .NET assembly (second container), which is packaged in a signed .msi file (third container).
using at least one control from said first control set to govern at least one aspect of use of said first protected information while said first protected information is contained within said first secure container;	The first secure container is a licensed and signed .NET component governed by the set of declarative statements comprising the LicenseProviderAttribute (one control).
creating a second secure container comprising a second control set for governing at least one aspect of use of protected information contained within said second secure container;	The second secure container is a .NET assembly, the protected information is the assembly and the second control set is declarative statement(s) within the assembly's header.
incorporating a first portion of said first protected information in said second secure container, said first portion made up of some or all of said first protected information;	Included in the .NET assembly is the .NET component.
using at least one control to govern at least one aspect of use of said first portion of said first protected information while said first portion is contained within said second secure container; and	The declarative statement(s) govern the use of the .NET component and the custom LicenseProvider class (first control set) controls the .NET component.
incorporating said second secure container containing said first portion of said first protected information within a third secure container comprising a third control set.	The third secure container is the signed .msi file in which the .NET assembly developer packages its assembly. The third control set is the conditional syntax statements written by the assembly developer and placed into the signed .msi file.

2.	Infringement is based on Microsoft's Visual Studio .NET and/or the .NET Framework licensing tools (in the.NET Framework SDK) and/or Microsoft Installer SDK..
A system for supporting electronic commerce including:	
means for creating a first secure control set at a first location;	The first location is a .NET component developer's site. The first secure control set is the set of declarative statements comprising the *LicenseProviderAttribute* of a first .NET licensed component that provides for a design-time license to use the control. This attribute also specifies the type of license validation that occurs. The component is encapsulated in a signed .NET assembly.
means for creating a second secure control set at a second location;	The second location is the .NET application developer's site where a .NET application comprising one or more assemblies is created. The second secure control set comprises the declarative statement(s) (including licensing statements, and code access security statements) of a signed .NET assembly using or calling the first .NET component. The control set can include a set of security permissions demanded by the .NET assembly containing the licensed component, whereby the permissions are demanded of components that call the application components. The control set can also be extended by controls expressed as conditional syntax statements in a signed .msi file containing a click through end-user license (the end-user license scenario).
means for securely communicating said first secure control set from said first location to said second location; and	The first .NET control set is securely communicated from the first location developer to the .NET solution provider by either being contained in a signed assembly, within a signed cabinet file or within a signed .msi file.
means at said second location for securely integrating said first and second control sets to produce at least a third control set comprising plural elements together comprising an electronic value chain extended agreement.	At the second location, the solution developer uses the .NET runtime that includes the LicenseManager. Whenever a class (control or component) is instantiated (here, an instance of the first .NET licensed component), the license manager accesses the proper validation mechanism for the control or component. A value chain is created through the creation of a run-time license for use of the first .NET component in the context of use of the .NET application developed at the second location. The

	license controls for the runtime license (derived from the design time license) are bound into the header of the .NET application assembly, along with the second control set. The creation of runtime license controls is securely handled by Visual Studio.NET or the LC tool. Runtime licenses are embedded into (and bound to) the executing assembly. The license control attribute included in the first .NET component is customized in the second location to express and require the runtime license. In a different scenario, the LC tool is used to create a ".licenses file" containing licenses for multiple components, including runtime licenses for components and classes created by the license provider. This .licenses file is embedded into the assembly. The third control set is an extended value chain agreement that comprises the runtime license controls for the first .NET licensed class (that had been bound to the assembly), the declarative controls provided by the solution provider in the solution provider's assembly, and any runtime licenses for other components included by the solution provider in the solution provider's assembly, and any end user license agreement provided by the application provider. The controls are typically integrated into the header of the .NET application assembly calling the first .NET licensed component. A further "end user licensing scenario" occurs when, at the second location, the application developer packages the application into a signed .msi file that includes conditional syntax statement controls that require that a user read and agree to an end user license agreement for the application and the embedded first component. The third control set includes a plurality of elements that include the run-time licenses mentioned above, security permissions controls, EULA controls (a fourth control set), all securely bound into the signed .msi file.

11. A system as in claim 2 in which said first location and said second location are contained within a Virtual Distribution Environment.	The Microsoft .NET Framework provides a Virtual Distribution Environment. Here the nodes are the Common Language Runtime instances that interpret the controls contained within .NET assemblies (among other functions).
29. A system as in claim 2 in which said first secure control set includes required	The licensing control in the first control set specifies the method required to validate

terms.	the license.
32. A system as in claim 2 in which said second secure control set includes required terms.	The security permissions demanded (as described above) are required terms for execution of the application code elements.
60. A system as in claim 2 in which said means for securely integrating said first and second control sets includes a fourth control set.	In the scenario where the application assembly is distributed using a signed .msi file, the secure integration of the first and second control sets is enhanced by the tamper protection afforded by the signed .msi file. In the end user license scenario, a fourth control set consisting of conditional syntax statements is included in the .msi file.
130. A system as in claim 2 further including means for executing said third control set within a protected processing environment.	The third control set is executed under the auspices of the CLR.
132. A system as in claim 130 in which said protected processing environment is located at a location other than said second location.	The third control set is executed at an end-user site within the CLR.
161. A system as in claim 2 in which said third control set includes controls containing human-language terms corresponding to at least certain of the machine-executable controls contained in said third control set.	In the end user license scenario, the third control set includes a fourth control set that requires that the human user agree with license terms displayed to the user. These human readable terms are referenced in the conditional syntax statement controls contained in the signed .msi file.
162. A method as in claim 161 in which said human-language terms are contained in one or more data descriptor data structures.	The .msi file is a data descriptor data structure.
170. A system as in claim 2 in which said means for creating a first secure control set includes a protected processing environment.	The creation of the first licensed component, including its licensed controls is carried out under the auspices of the CLR.
171. A system as in claim 2 in which said means for creating a second secure control set includes a protected processing environment.	The application design time environment and the creation of the .NET application is carried out under the auspices of the CLR.
172. A system as in claim 2 in which said means at said second location for securely integrating includes a protected processing environment.	The means for integrating the runtime license with the application controls is carried out under the auspices of the CLR.
329. A system as in claim 2 in which said	VS.NET runs under Windows.

means for creating a first secure control set includes an operating system based on or compatible with Microsoft Windows.	
330. A system as in claim 2 in which said means for creating a second secure control set includes an operating system based on or compatible with Microsoft Windows.	VS.NET runs under Windows.
331. A system as in claim 2 in which said means at said second location for securely integrating said first and second control sets includes an operating system based on or compatible with Microsoft Windows.	VS.NET runs under Windows.
346. A system as in claim 2 further comprising means by which said third control set governs the execution of at least one load module.	The third control set in the scenario described in the claim map for claim 2 governs a portable .NET executable designed to be loaded into the CLR environment (a CLR host).
347. A system as in claim 2 farther comprising means by which said third control set governs the execution of at least one method.	The third control set in the scenario described in the claim map for claim 2 governs a .NET executable. This executable contains one or more methods.
349. A system as in claim 2 further comprising means by which said third control set governs the execution of at least one procedure.	The third control set in the scenario described in the claim map for claim 2 governs a .NET executable. This executable contains one or more procedures.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,112,181

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
48.	Infringing products include Microsoft SMS (Systems Management Server) 2.0 and subsequent versions.
A method for narrowcasting selected digital information to specified recipients, including:	
a) at a receiving appliance, receiving selected digital information from a sending appliance remote from the receiving appliance,	The *receiving appliance* is the client (e.g., end user computer in an Enterprise setting) receiving *digital information* (packages and/or advertisement files) from the *sending appliance*, the centralized SMS database via a Client Access Point and/or Distribution Point set up on a server.
the receiving appliance having a secure node and being associated with a specified recipient;	The "node" is "secure" as a result of SMS security, as well as how it identifies and selects clients. The "specified recipient" is the result of the *collection* identifying a specific client that meets the criteria for a package or advertisement.
i) the digital information having been selected at least in part based on the digital information's membership in a first class, wherein the first class membership was determined at least in part using rights management information; and	The *digital information* is a software package or advertisement. The "*first class membership was determined in part using rights management information*" reads on creating software packages (or advertisements) based on attributes of the software.
ii) the specified recipient having been selected at least in part based on membership in a second class, wherein the second class membership was determined at least in part on the basis of information derived from the specified recipient's creation, use of, or interaction with rights management information; and	The "specified recipient" is the client selected to receive a package or advertisement. That recipient is chosen based on a collection rule, or on the recipient's possession of a license.
b) the specified recipient using the receiving appliance to access the received selected digital information in accordance with rules and controls, associated with the selected digital information.	The *receiving appliance* is the client computer. The SMS agents on the client computer receive, evaluate and take the appropriate action based on *rules and controls* governing the package and/or advertisement (i.e. the *selected digital information*).
the rules and controls being enforced	Rules and controls are enforced by Agents on

by the receiving appliance secure node.	the client (the *secure node*)
59. The method of claim 48 wherein said received selected digital information is at least in part event information.	Event information includes SMS event information, including *Scheduling Classes* .
63. The method of claim 48 wherein said received selected digital information is at least in part executable software.	All SMS packages must include a minimum of one program.
70. The method of claim 48 wherein said rules and controls at least in part govern usage audit record creation.	A control governs whether a MIF (management information file) is sent back to the SMS db after installation is done to report on the success or failure of the installation.
89. The method of claim 48 wherein said receiving appliance is a personal computer.	The primary purpose of SMS is to manage software on personal computers throughout the Enterprise.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,112,181

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
48.	Infringing products include Windows Media Player and Windows Media Rights Manager
A method for narrowcasting selected digital information to specified recipients, including:	This claim pertains to Windows Media Player with Individualized DRM Client and Windows Media Rights Manager used in the context of a narrowcast pay-per-view (hear) media distribution service., simulcast and/or subscription services.
(a) at a receiving appliance, receiving selected digital information from a sending appliance remote from the receiving appliance, the receiving appliance having a secure node and being associated with a specified recipient	Receiving appliance is a user's PC with individualized DRM client (secure node). Specified recipient is a user using the specific individualized DRM client to access and render narrowcast pay-per-view media, simulcast and/or subscription services for which the user acquires a license.
(i) the digital information having been selected at least in part based on the digital information's membership in a first class, wherein the first class membership was determined at least in part using rights management information; and	The digital information is media that is narrowcast to licensed recipients. These narrowcast streams are licensed to users who have acquired licenses and whose PCs (appliances) support WMPs that have individualized DRM clients. This attribute is included in the signed WMA file header and is used in the process of acquiring licenses for access to the media. Media that are licensed to the recipient have their licenses bound to the recipient's Individualization module.
(ii) the specified recipient having been selected at least in part based on membership in a second class, wherein the second class membership was determined at least in part on the basis of information derived from the specified recipient's creation, use of, or interaction with rights management information; and	The recipient is selected for this content based on the fact that the recipient is a member of the class of recipients who have a license for the narrowcast media and whose devices support WMP and individualized DRM clients. The recipient's machine must indicate support for individualization in challenges that are sent as part of requests for media in this narrowcast class.
(b) the specified recipient using the receiving appliance to access the received selected digital information in accordance with rules and controls associated with the selected digital information, the rules and controls being enforced by the receiving appliance secure node.	Recipient's machine uses WMP and the individualized DRM client to access the narrowcast media in accordance with all rules associated with the media and contained in the media license – in particular, requirements that individualization be supported.

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
61. The method of claim 48 wherein said received selected digital information is at least in part entertainment information.	The digital information is Windows Media, which encodes audio/visual entertainment content.
62. The method of claim 61 wherein said entertainment information is at least in part music information.	Reads on narrowcast Windows Media Files that are music or audio/visual.
67. The method of claim 48 wherein said rules and controls at least in part use digital certificate information.	The license contains a digital certificate. The DRM client uses the certificate in the license to verify this signature and to verify that the header has not been tampered with.
72. The method of claim 48 wherein said rules and controls in part specifying at least one clearinghouse acceptable to rightsholders.	The signed header contains at least one URL that indicates to the Windows Media Rights Manager the license clearinghouse to be used in acquiring licenses.
75. The method of claim 72 wherein said at least one acceptable clearinghouse is a rights and permissions clearinghouse.	This clearinghouse is a license clearinghouse responsible for mapping rights and permissions onto requested content or narrowcasts and binding them to the requesting client environment or user of this environment.
89. The method of claim 48 wherein said receiving appliance is a personal computer.	Windows Media Player and the Individualized DRM client run on a personal computer.

91	Infringing products include Windows Media Player and Windows Media Rights Manager
A method for securely narrowcasting selected digital information to specified recipients including:	This claim pertains to Windows Media Player with Individualized.DRM Client and Windows Media Rights Manager used in the context of a narrowcast simulcast, pay-per-view (hear) media distribution service. and/or subscription services. The content is delivered in a Protected Windows Media File.
(a) receiving selected digital information in a secure container at a receiving appliance remote from a sending appliance, the receiving appliance having a secure node, the receiving appliance being associated with a receiving entity	Narrowcast content is received in a Protected Windows Media File. Receiving appliance is user's PC with individualized DRM client (secure node).
(i) the digital information having been selected at least in part based on the digital information's membership in a first class,	The digital information is media that is narrowcast to licensed recipients (for example, a sold-out concert is narrowcast on the Internet to "the class of" licensed (or ticketed) viewers).
(ii) the first class membership having been determined at least in part using rights management information	These narrowcast streams are licensed to users who have acquired licenses and whose PCs (appliances) support WMPs that have individualized DRM clients. This attribute is included in the signed WMA file header and is used in the process of acquiring licenses for access to the media. Media that are licensed to the recipient have their licenses bound to the recipient's individualization module.
(b) the receiving entity having been selected at least in part based on said receiving entity's membership in a second class,	The recipient is selected for this content based on the fact that the recipient is a member of the class of recipients who has a license for the narrowcast media.
(i) the second class membership having been determined at least in part on the basis of information derived from the recipient entity's creation, use of, or interaction with rights management information	The recipient class is determined by the license bound to the user's device that supports WMP and individualized DRM clients. The recipient's machine must indicate support for individualization in challenges that are sent as part of requests for media in this narrowcast class.
(c) receiving at the receiving appliance rules and controls in a secure container,	Receives a protected Windows Media File
(i) the rules and controls having been associated with the selected digital information; and	Receives a license that is bound to the file as well as to the specific DRM client individualization information.
(d) using at the receiving appliance the selected digital information in accordance	Recipient's machine uses WMP and the individualized DRM client to access the

with the rules and controls,	narrowcast media in accordance with all rules associated with the media and contained in the media license – in particular, requirements that individualization be supported.
(i) the rules and controls being enforced by the receiving appliance secure node.	The WMP and DRM client enforce the rules embedded in the Protected Windows Media File License.
104. The method of claim 91 wherein said received selected digital information includes entertainment information.	The digital information is Windows Media, which encodes audio/visual entertainment content.
109. The method of claim 91 wherein said rules and controls at least in part use digital certificate information.	The license contains a digital certificate. The DRM client uses the certificate in the license to verify this signature and to verify that the header has not been tampered with.
114. The method of claim 91 wherein said rules and controls specify at least one clearinghouse acceptable to rightsholders.	The signed header contains at least one URL that indicates to the Windows Media Rights Manager the license clearinghouse to be used in acquiring licenses.
117. The method of claim 114 wherein said at least one acceptable clearinghouse is a rights and permissions clearinghouse.	This clearinghouse is a license clearinghouse responsible for mapping rights and permissions onto requested content or narrowcasts and binding them to the requesting client environment or user of this environment.
131. The method of claim 91 wherein said receiving appliance is a personal computer.	Windows Media Player and the individualized DRM client run on a personal computer.

INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.
INTERTRUST INFRINGEMENT CHART
FOR U.S. PATENT NO. 6,389,402

CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
1.	Products infringing: Microsoft Visual Studio .NET, .NET License Compiler, .NET Framework SDK, and .NET Common Language Runtime
A method including	A method for producing a third .NET component (application) that incorporates first and second .NET component whose distribution is license controlled.
creating a first secure container including a first governed item and having associated a first control;	The *first secure container* is a first signed .NET component that includes a license control. The *governed item* is the .NET component. The *first control* is the set of declarative statements comprising the LicenseProviderAttribute of a first .NET licensed component that provides for a design-time license to use the control. This attribute also specifies the type of license validation that occurs.
creating a second secure container including a second governed item and having associated a second control;	The *second secure container* is the second signed .NET component that includes a license control. The *governed item* is the .NET component. The *second control* is the set of declarative statements comprising the LicenseProviderAttribute of a second .NET licensed component that provides for a design-time license to use the control. This attribute also specifies the type of license validation that occurs.
transferring the first secure container from a first location to a second location;	The creator distributes a signed and licensed .NET component. An application developer at a second location downloads a first .NET component for inclusion into an application.
transferring the second secure container from a third location to the second location;	A creator distributes a signed and licensed .NET component from a different location. Application developer downloads a second .NET component for inclusion into an application.

at the second location, obtaining access to at least a portion of the first governed item, the access being governed at least in part by the first control;	At the *second location*, the application developer uses the .NET runtime that includes the LicenseManager to access a *first governed item.* Whenever a class (control or component) is instantiated (here, an instance of the first .NET licensed component), the license manager accesses the proper validation mechanism for the control or component. The *first control* comprises the declarative statement(s) (including licensing statements, and code access security statements) of the first .NET component.
at the second location, obtaining access to at least a portion of the second governed item, the access being governed at least in part by the second control;	At the *second location*, the application developer uses the .NET runtime that includes the LicenseManager to access a *second governed item.* Whenever a class (control or component) is instantiated (here, an instance of the second .NET licensed component), the license manager accesses the proper validation mechanism for the control or component. The *second control* comprises the declarative statement(s) (including licensing statements, and code access security statements) of the second .NET component.
at the second location, creating a third secure container including at least a portion of the first governed item and at least a portion of the second governed item and having associated at least one control, the creation being governed at least in part by the first control and the second control.	At the *second location*, the application developer uses the .NET runtime that includes the LicenseManager to access a *first governed item* and *second governed item* to construct an application, the *third secure container.* *Creation governance* is accomplished by invoking the .NET runtime to access the *first governed item* and the *second governed item.* Whenever a class (control or component) is instantiated the license manager accesses the proper validation mechanism for the control or component. The *portions* of the first governed item and second governed item that are being included in the third secure container will typically include the governed items themselves, ie. the .NET components. The *associated control* in this case is the LicenseProviderAttribute, created and inserted into the application.

EXHIBIT C

CONFIDENTIAL—SUBJECT TO PROTECTIVE ORDER OF NOVEMBER 19, 2001: Exhibit C contains documents or things that are the subject of a Protective Order of this Court and cannot be opened or its contents made available to anyone other than this Court or counsel of record for the parties.

WILLIAM L. ANTHONY (State Bar No. 106908)
ERIC L. WESENBERG (State Bar No. 139696)
HEIDI L. KEEFE (State Bar No. 178960)
BAS DE BLANK (State Bar No. 191487)
ORRICK, HERRINGTON & SUTCLIFFE, LLP
1000 Marsh Road
Menlo Park, CA 94025
Telephone: (650) 614-7400
Facsimile: (650) 614-7401

STEVEN ALEXANDER (admitted *Pro Hac Vice*)
JAMES E. GERINGER (admitted *Pro Hac Vice*)
JOHN D. VANDENBERG
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, OR 97204
Telephone: (503) 226-7391
Facsimile: (503) 228-9446

Attorneys for Defendant and Counterclaimant,
MICROSOFT CORPORATION

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

OAKLAND DIVISION

INTERTRUST TECHNOLOGIES CORPORATION, a Delaware corporation,

Plaintiff,

v.

MICROSOFT CORPORATION, a Washington corporation,

Defendant.

AND RELATED CROSS-ACTION.

Case No. C 01-1640 SBA (MEJ)

Consolidated with C 02-0647 SBA (MEJ)

DEFENDANT MICROSOFT CORPORATION'S PRELIMINARY INVALIDITY CONTENTIONS

(Patent Local Rules 3-3 and 3-4)

I. Patent Local Rule 3-3(a) Identification of Prior Art

Pursuant to Patent Local Rule 3-3, Defendant Microsoft Corporation ("Microsoft") makes the following Preliminary Invalidity Contentions[1] with respect to the following patents asserted by plaintiff InterTrust Technologies Corporation ("InterTrust") in this action: U.S. Patent No. 6,185,683 ("the `683 patent"); U.S. Patent No. 6,253,193 ("the `193 patent"); U.S. Patent No. 5,920,861 ("the `861 patent"); U.S. Patent No. 5,982,891 ("the `891 patent"); U.S. Patent No. 5,917,912 ("the `912 patent"); U.S. Patent No. 6,157,721 ("the `721 patent"); U.S. Patent No. 5,915,019 ("the `019 patent"); U.S. Patent No. 5,949,876 ("the `876 patent"); U.S. Patent No. 6,112,181 ("the `181 patent"); and U.S. Patent No. 6,389,402 ("the `402 patent").

Despite the length of time this case has been pending, discovery is still at an early stage due to intervening stays. InterTrust continues to assert eleven patents and over one hundred and fifty claims. In view of these factors, Microsoft continues to evaluate the prior art at this time. Microsoft reserves the right to amend or supplement its Preliminary Invalidity Contentions to take into account prior art, information or defenses that might come to light as a result of its continuing discovery efforts, errors subsequently recognized by any of the parties, and as a result of further evaluation of the prior art.[2] In addition, Microsoft has moved to strike InterTrust's September 2, 2003 PLR 3-1 Preliminary Infringement Contentions as being insufficient. To the extent that the Court grants Microsoft's motion and orders InterTrust to amend/re-serve its 3-1 statement in compliance with the Local Rules, Microsoft reserves the right to amend or supplement its PLR 3-3 Preliminary Invalidity Contentions in response to any amended infringement contentions submitted by InterTrust. Microsoft further reserves the right to rely

---

[1] These Preliminary Invalidity Contentions incorporate by reference Microsoft's prior Preliminary Invalidity Contentions dated August 7 and 16, 2002.

[2] For example, Microsoft reserves the right to amend/supplement this disclosure once InterTrust complies with discovery responses, which Microsoft contends are incomplete and inadequate. To date, Microsoft has objected to InterTrust's continued refusal to provide information sought in discovery, including, but not limited to: the identity of the alleged inventors of specific claims; conception or actual reduction to practice dates for specific claims; whether to there has ever been any alleged embodiment(s) of the asserted claims; and what, if any, specification support is alleged, including from any of the applications for which InterTrust claims priority. Each of these pieces of information could affect the priority date for any given claim, expanding or narrowing the window of applicable prior art. Without this information, which is within InterTrust's exclusive knowledge and control, Microsoft's PLR 3-3 Contentions are subject to amendment and/or supplementation.

upon InterTrust's own activities, alone and in connection with others. Microsoft further reserves the right to amend this statement or otherwise further respond if InterTrust contends (or the Court rules) that any earlier or later priority dates may apply for individual claims. Microsoft also reserves its right to amend or supplement these invalidity contentions pursuant to Patent Local Rule 3-6 and 3-7.

Attached hereto, as Appendix A, is a listing showing "the identity of each item of prior art that allegedly anticipates each asserted claim or renders it obvious" (PLR 3-3(a)). On information and belief, each listed publication became prior art at least as early as the dates given. In addition, the citations and explanations provided in the exhibits are mere examples, and Microsoft reserves its right to rely on any other portions or aspects of the prior art references and systems that may also disclose or practice elements of the asserted claims. Patent Local Rule 3-3 does not require identification of evidence that establishes the inherence of a claim element in an item of prior art, nor does it require identification of evidence that establishes knowledge of those of ordinary skill in the relevant fields of art. Accordingly, Microsoft does not purport to have provided all such information in the attached exhibits.

From InterTrust's current document production, it appears that its employees' and consultants' activities, including offers for sale, public uses, derivations, "inventions" (as the word is used in 35 U.S.C. § 102(g)), and disclosures to Willis Ware, Drew Dean, and others not under any duty of confidentiality, constituted or created material and perhaps anticipatory prior art to many of the asserted claims. This art was not cited to the Patent Office. Discovery is ongoing, and Microsoft reserves the right to amend or supplement this disclosure after Microsoft has had an opportunity to investigate this possible prior art during discovery.

II. Patent Local Rule 3-3(b) and 3-3 (c) Classification and Analysis of Prior Art

Microsoft contends that at least one term or phrase in each of the asserted claims is indefinite under 35 U.S.C. § 112, and hence, each of the asserted claims is incapable of construction. However, for the limited purpose of classification and analysis of prior art, Microsoft has construed the claim terms in a manner consistent with the apparent construction of terms offered by InterTrust in its Revised Preliminary Infringement Contentions. **Microsoft does**

not agree with these constructions, and nothing in these Preliminary Invalidity Contentions should be construed as an admission, a declaration against interest, whether under the Federal Rules of Evidence or otherwise, as to what a particular claim limitation means. For this reason, Microsoft's identification of "corresponding structures" for "means-plus-function" limitations that are set out in the Preliminary Invalidity Charts are not admissions as to the identity of such structures. Rather, they are based upon Microsoft's best guess as to what InterTrust may someday identify as corresponding structures for the means-plus-function limitations of its asserted claims, to the extent that Microsoft understands them.[3]

Accordingly, Microsoft's Preliminary Invalidity Contentions should not be construed as advocating a particular claim construction for any disputed claim terms. For the limited purpose of providing Preliminary Invalidity Contentions, and subject to the conditions set forth above, Microsoft has, to the extent possible, attempted to construe the claims in a manner consistent with InterTrust's Revised Preliminary Infringement Contentions.

Pursuant to Patent Local Rules 3-3(b) and 3-3(c), Microsoft provides the classification of prior art in the listing and charts attached hereto as Appendices A and B. Appendix A, beyond identifying each item of prior art, further indicates whether each prior art reference is used as an anticipatory reference and/or as a reference which, alone, or in combination with other prior art, renders the claims obvious. Appendix B includes charts which (1) specifically identify where in each item of prior art each element of each asserted claim is found and (2) establish how that prior art anticipates or renders obvious all of the asserted claims. In the event that any charted prior art is found not to be anticipatory under 35 U.S.C. § 102, Microsoft reserves the right to rely upon that art to prove obviousness under 35 U.S.C. § 103. Likewise, in the event InterTrust

[3] To date, InterTrust has refused to identify any structure corresponding to the means-plus-function elements in its asserted claims. It is Microsoft's position that this is a violation of the Patent Local Rules, and that as a result of refusing to identify a structure associated with each means-plus-function element, InterTrust admits that there is no such structure disclosed, has waived its right to assert claimed structure, and that those claims are therefore invalid at least for failure to satisfy the written description requirement of 35 U.S.C. §112. *See* InterTrust's Patent Local Rule 3-1 served September 2, 2003 and InterTrust's Opposition to Microsoft's Motion to Strike InterTrust's PLR 3-1 Contentions.

amends or supplements its Preliminary Infringement Contentions, Microsoft reserves its rights to amend and supplement its Preliminary Invalidity Contentions.

To the extent that any prior art produced to InterTrust has not been classified as prior art under 35 U.S.C. §§ 102 or 103, Microsoft reserves the right to rely on this art or supplement its disclosure for the following reasons:

(i) Microsoft's position on the invalidity of particular claims will depend on how those claims are construed by the Court. As thus far only preliminary claim construction has occurred Microsoft cannot take a final position for the bases for invalidity of disputed claims. The Court's subsequent claim constructions of remaining terms may yield constructions different from what Microsoft assumes herein.

(ii) Microsoft is continuing to diligently search for relevant prior art but has not yet completed that search and continues to evaluate prior art that has been located.

(iii) Microsoft has not completed its discovery from Plaintiff or from third parties with knowledge of the relevant prior art. Depositions of the persons involved in the drafting and prosecution of the patents-in-suit, the inventors, and persons who attempted to practice InterTrust's claimed invention, for example, will likely affect Microsoft's contentions.

A. Prior Art Under 35 U.S.C. § 102 Which Anticipates The Asserted Claims of Each of the Asserted Patents

Subject to the above-referenced qualifications concerning the preliminary nature of this disclosure, Microsoft believes a reasonable basis exists that, as more particularly explained in the Preliminary Invalidity Contentions attached as Appendix B hereto, the references listed in Appendix B anticipate the asserted claims of the each of the asserted patents.

B. Prior Art Under 35 U.S.C. § 103 Which Renders Obvious One or More of the Asserted Claims

Each of the references called out in Appendix A can be combined with one another so as to render one or more of the claims of the asserted patents invalid as obvious; and many of them are explicitly motivated to do so by virtue of extensive cross-references to one another's solutions. InterTrust is currently asserting 151 claims in eleven patents, which cite hundreds of references. Hundreds of additional non-cited relevant prior art has been uncovered and cited to

InterTrust. The number of potential combinations of these references, if only two or a few references are combined for each claim, is necessarily very large. Microsoft requests InterTrust to reduce its asserted claims so as to reduce the number of combinations to a manageable number. Nonetheless, Microsoft has provided mapping of combinations as discussed below. Indeed, even where explicit cross-referencing and incorporation by reference does not exist, the motivation to combine any of the references arises from the common objectives and subject matter, digital rights management. The common objectives and subject matter are expressed generally in the claim charts of Appendix B, which are incorporated by reference into Microsoft's showing under 35 U.S.C. § 103.

The motivation for seeking "security," privacy and integrity was widely recognized in the United States and elsewhere prior to February 13, 1994, and since prior to February 13, 1994, has extended to any information or item of perceived value, including books, music, games, computer systems, other computer programs, and any digital data or content that maybe deemed valuable or worthy of protection. Additional motivations to combine references include the desire to meet or exceed any applicable laws or industry or government standards, such as the Orange Book, Computer Fraud and Abuse Act of 1986, Computer Security Act of 1989 PL100-35, High Performance Computing Act (HPCA) of 1991 (PL102-194), and 17 U.S.C. §§ 101 et seq. Industry standards include those for communication such as X.509, TCP/IP, WWW, and WAIS, and those for encryption or transmission of encrypted information, *e.g.* DES, Triple DES, RSA, SSL, MIME, S/MIME, SHTTP, HTTPS, MD5, and PEM. Additional teachings to combine these references with such items of information include "security" (including "security" levels), permissions, certificates, tickets, "secure" processors, "secure" storage, "smart" cards (including smart cards able to store data and perform computations such as encryption/decryption), tamper resistance techniques for hardware and software, physical "security", and "trusted" time. Also included are authentication and authorization in trusted distributed systems, enabling software or features thereof to run only on particular machines or in particular ways, and treating binary information/data at varied levels of granularity.

It was further obvious to combine any of these "security" features with any of the software or hardware available at the time. For example, it would have been obvious to combine any file and operating systems such as NT, NFS, Andrew, Netware, Mach, DT Mach, Multics, Amoeba, ISOS, and Unix; or protocols, codes and systems such as secure kernels, WWW, SSL, SGML, hyptertext, Oak, Telescript, OOP and other programming technologies or frameworks (*e.g.* Smalltalk, COM, OLE, Bento, OpenDoc; object oriented databases with watermarking; obfuscation; swIPe; SNMP; auditing; on-line (or other digitally transmitted) transaction and subscription-based services and billings; electronic payment; on-line banking, entertainment and commercial interactive commerce; ATMs; encryption and authentication; physical security tools and devices; physically secure locations; physically "secure" products such as tamper resistant computer or other devices, "secure" processors, "secure" memory, "smart" cards, set-top boxes, portable devices, "secure" communications facilities, electronic wallets.[4]

III. Patent Local Rule 3-3(d) Disclosure: Invalidity For Failure to Satisfy 35 U.S.C. § 112.

Each of the asserted InterTrust patent claims is invalid as indefinite, for inadequate written description and for lack of enablement as those requirement are set forth by 35 U.S.C. § 112.[5] In accordance with Patent L.R. 3-3(d), Microsoft identifies in Appendix C, attached hereto, exemplary bases, on an element by element basis, for invalidating each asserted claim of each asserted patent for indefiniteness and lack of an adequate written description. The asserted claims are unclear in scope and not nearly as precise as the subject matter allows.

Appendix C contains examples of why the indefiniteness prohibited by 35 U.S.C. § 112(2) arises from many causes, including:

a) use of terms that lack an ordinary meaning in the art and are undefined in the specification;

---

[4] These examples are not intended to be an exhaustive list and are set forth for illustrative purposes.

[5] Microsoft also asserts that one or more of the claims are invalid under 35 U.S.C. § 112(1) for failure to identify the "best mode" for carrying out the invention. However, pursuant to Patent L.R. 3-3(d), Microsoft's arguments related to that defense are not required to be set forth in the attached charts, and hence are not included in Exhibit C.

b) use of terms that are used in the specification in a manner which is internally inconsistent, as well as inconsistent with their ordinary meaning, but are not specifically defined in the specification;

c) InterTrust's refusal to identify the structure in the application's written description linked to claim elements subject to 35 U.S.C. § 112, ¶6 ("means (or step) plus function);

d) such excessive disclaimers of specificity of a term that the term becomes meaningless;

e) inconsistent uses of a term within a single specification;

f) inconsistent uses of a term between a specification and something allegedly incorporated into that specification;

g) inconsistencies within the language of a given claim;

h) inclusion of the same element twice in a claim, resulting in improper double inclusion of an element;

i) impermissible reference to trademarks in a claim;

j) inconsistent use of terms that may be synonyms for one another or that could be used to mean same thing or different things.

The indefiniteness of the asserted claims is exacerbated by InterTrust's attempt to apply these claims to the very different structures and techniques of (or those that InterTrust wrongly attributes to) the Microsoft accused products. Microsoft reserves the right to modify this listing, *e.g.*, if and when InterTrust clarifies its infringement contentions and claim construction positions.

Appendix C also provides examples of the lack of an adequate written description supporting the asserted claims. For example, the asserted claims fail for lack of an adequate written description under 35 U.S.C. § 112(1) to the extent that they are construed to contradict and/or fail to require the essential, non-optional alleged attributes of the alleged "inventions" identified in their specifications (and any specification allegedly incorporated by reference) and the applications from which the patents issued. The asserted claims also fail to comply with the

written description requirement as set forth in *Gentry Gallery, Inc v. Berkline Corp.*, 134 F.3d 1473 (Fed. Cir 1998) to the extent that the scope of any of them exceeds the scope of the alleged "invention" as set forth in the accompanying specification (and any specification allegedly incorporated therein). For example, in the specification of U.S. Patent No. 6,253,193 InterTrust states that:

> The present invention assertedly provides a new kind of "virtual distribution environment" (called "VDE" in this document) that secures, administers, and audits electronic information use. VDE also features fundamentally important capabilities for managing content that travels "across" the "information highway." These capabilities comprise a rights protection solution that serves all electronic community members. These members include content creators and distributors, financial service providers, end-users, and others. VDE is the first general purpose, configurable, transaction control/rights protection solution for users of computers, other electronic appliances, networks, and the information highway.

Accordingly any claims that rely on this specification must be limited in scope to the invention described therein. To the extent that they exceed the scope of what is described, they are invalid under the written description requirement.

Microsoft further contends that each asserted claim, when viewed in its entirety, is invalid under 35 U.S.C. § 112(1) because the specifications of the patents fail to teach one of ordinary skill in the art how to practice the entirety of the broad scope of those claims without undue experimentation.

For example, based on the specification, most if not all of the claims involve the use of software of one kind or another, yet the specification does not disclose any software programs that could be used or adapted for use in practicing the claimed inventions. In addition to failing to disclose any software program by explicit reference, the patent specifications does not describe with sufficient specificity the identity of software programs needed to practice the claimed invention that would prevent the need for undue experimentation by a person skilled in the art to practice the claimed inventions. The claims set forth a multiplicity of functions, features, and characteristics for the purported inventions, and the specifications are replete with references to software necessary to practicing the inventions, yet the specification neither identifies enabling software that satisfies such requirements, nor provides guidance that would

allow a person of ordinary skill in the art to program enabling software without undue experimentation.[6]

As shown in Appendix C[7], asserted claims contain terms that are subject to multiple definitions, and the patent specifications do not disclose one or more of the alternate definitions. The full scope of the claim is therefore not described or taught in the specification. Any claim in Appendix C that contains a claim term subject to multiple definitions fails to teach the full scope of the claim and therefore fails the enablement requirement if the specification does not specify the operative definition for the term.

There are numerous other reasons that the unprecedented breadth of scope of the claims asserted by InterTrust are not enabled, including InterTrust's failure to implement the claims after substantial investment of time, labor, and money. Given the complexity of the asserted patents and their interdisciplinary subject matter, the state of the prior art, the absence of predictability of the prior art, the amount of experimentation necessary to practice the patents, the absence of embodiments, and the absence of guidance for practicing the invention provided in the specification[8], the relative skill of those practicing the art and the breadth of the claims, the asserted claims fail to meet the enablement requirement of 35 U.S.C. § 112 ¶ 1.

The full claims of the asserted patents fail to satisfy the enablement and written description requirements for the following reasons:

The '683 Patent

Claim 2: Claim 2 of the '683 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling

---

[6] In its discovery responses, InterTrust refuses to identify software programs necessary for practicing the inventions purportedly disclosed in the asserted patents. *See* InterTrust responses to Microsoft Interrogatory Nos. 39 and 40.

[7] *See* Appendix C for further element by element analysis of invalidity for failure to satisfy 35 U.S.C. § 112 ¶ 1. The indefiniteness of the claim terms addressed in Exhibit C affect enablement because the indefiniteness of the claim terms prevents the specification from adequately teaching a person of skill in the art how to make and use the full scope of the claimed inventions without undue experimentation.

[8] The failure of the specifications to provide necessary guidance also establishes that the claims fail to meet the written description requirement of 35 U.S.C. § 112 ¶ 1.

software and operation of such software on accompanying hardware. Specifically, limitations in Claim 2 (63:40-66), both explicitly and implicitly require software. Since no software is disclosed in the specification, and since the specification provides no useful programming guidance, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 2. Claim 2 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "security", "secure container," "containing"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 2 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 3: Claim 3 of the '683 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several limitations in Claim 3 (64:6-30), both explicitly and implicitly require software. Since no software is disclosed in the specification, and insufficient programming guidance (if any) is provided by the specification, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 3. Claim 3 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "security", "secure container," "rule"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 3 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 4: Claim 4 is dependent upon Claim 3 and thus fails the enablement and

written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 4 fails because it requires additional undisclosed software.

Claim 5: Claim 5 of the '683 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several limitations in Claim 5 (64:41-66), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 5. Claim 5 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "security", "secure container," "governed item"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 5 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 6: Claim 6 is dependent upon Claim 5 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 6 fails because it requires additional undisclosed software..

Claim 28: Claim 28 of the '683 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several limitations in Claim 28 (70:20-59), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 28. Claim 28

also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "security," "electronic intermediary," "being associated with . . ."). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 28 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 29: Claim 29 is dependent upon Claim 28 and fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 29 fails because it requires additional undisclosed software. Claim 29 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "operatively connected"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed

Claim 56: Claim 56 of the '683 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several limitations in Claim 56 (77:34-56), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 56. Claim 56 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "security," "secure container," "secure electronic container"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated

above with respect to all of the claims, Claim 56 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 126: Claim 126 of the '683 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several limitations in Claim 126 (82:50-83:7), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 126. Claim 126 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "security," "secure digital container," "trusted intermediary services"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 126 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 127: Claim 127 is dependent upon Claim 126 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 127 fails because it requires additional undisclosed software. Claim 127 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "at least in part identifies"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed

The '193 Patent

Claim 1: Claim 1 of the '193 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the

purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several limitations in Claim 1 (320:62-321:18), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 1. Claim 1 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "budget control," "secure database," "copy control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 1 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 2: Claim 2 is dependent upon Claim 1 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 2 fails because it requires additional undisclosed software. Claim 127 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "a time substantially contemporaneous"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed

Claim 3: Claim 3 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 3 fails because it requires additional undisclosed software. Claim 3 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "encumbrance on said budget"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the

full scope claimed.

Claim 4: Claim 4 is dependent upon Claim 3 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 4 fails because it requires additional undisclosed software. Claim 4 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "digital file authorized by said budget"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 11: Claim 11 of the '193 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several limitations in Claim 11 (322:22-45), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 11. Claim 11 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "security," "secure memory," "features"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 11 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 15: Claim 15 of the '193 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several

limitations in Claim 15 (323:15-41), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 15. Claim 15 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "security," "secure database"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 15 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 16: Claim 16 is dependent upon Claim 15 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 16 fails because it requires additional undisclosed software. Claim 16 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "authentication step"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed

Claim 19: Claim 19 of the '193 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several limitations in Claim 19 (324:9-37), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 19. Claim 19 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.*

"clearinghouse"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 19 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 51: Claim 51 of the '193 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software and operation of such software on accompanying hardware. Specifically, several limitations in Claim 51 (326:51-327:12), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 51. Claim 51 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "security," "clearinghouse," "location remote from"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 51 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

The '861 Patent

Claim 34: Claim 34 of the '861 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 34 (24:65-25:15), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make

and use the full scope of Claim 34. Claim 34 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "descriptive data structure," "element information," "metadata rules"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 34 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 35: Claim 35 is dependent on Claim 34 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 35 fails because it requires additional undisclosed software. Claim 35 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "rights management data structure"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 36: Claim 36 is dependent on Claim 35 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 36 fails because it requires additional undisclosed software. Claim 36 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "content," "rules at least in part governing . . ."). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 37: Claim 37 is dependent on Claim 36 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 37 fails because it requires additional undisclosed software. Claim 37 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "descriptive data structure is stored within said first secure container"). The specification does

not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 44: Claim 44 is dependent on Claim 34 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 44 fails because it requires additional undisclosed software. Claim 44 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "representation of the format of data . . ."). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 45: Claim 45 is dependent on Claim 44 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 45 fails because it requires additional undisclosed software. Claim 45 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "information regarding elements . . ."). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 46: Claim 46 is dependent on Claim 44 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 46 fails because it requires additional undisclosed software. Claim 46 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "target data block"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 47: Claim 47 is dependent on Claim 46 and thus fails the enablement and

written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 47 fails because it requires additional undisclosed software. Claim 47 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "target data block," "target environment"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 48: Claim 48 is dependent on Claim 46 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 48 fails because it requires additional undisclosed software. Claim 48 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "source," "source message field"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 58: Claim 34 of the '861 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 34 (24:65-25:15), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 34. Claim 34 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "metadata information," "generating or identifying at least one rule . . ."). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims,

Claim 34 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 64: Claim 64 is dependent on Claim 58 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 64 fails because it requires additional undisclosed software. Claim 64 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "creation of said first secure container"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 67: Claim 67 is dependent on Claim 64 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 67 fails because it requires additional undisclosed software. Claim 67 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 68: Claim 68 is dependent on Claim 67 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 68 fails because it requires additional undisclosed software. Claim 68 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 71: Claim 71 is dependent on Claim 58 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 71 fails because it requires additional undisclosed software. Claim 71 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of

the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 72: Claim 72 depends to Claim 58 and fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 72 fails because it requires additional undisclosed software.

The '891 Patent

Claim 1: Claim 1 of the '891 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 1 (318:59-319:8), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 1. Claim 1 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "securely receiving," "secure operating environment," "control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 1 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 22: Claim 22 of the '891 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 22 (320:15-31) both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 22. Claim 22 also fails the enablement requirement in light of the breadth

of the subject matter claimed (*e.g.* "securely combining," "control arrangement," "securely requiring"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 22 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 23: Claim 23 is dependent on Claim 34 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 23 fails because it requires additional undisclosed software.

Claim 26: Claim 26 of the '891 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 26 (320:40-55) both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 26. Claim 26 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "composite data item," securely providing,"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 26 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 27: Claim 27 is dependent on Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 27 fails because it requires additional undisclosed software. Claim 27 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "combining step"). The specification does not teach a person of ordinary skill in the art how to

practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 28: Claim 28 is dependent on Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 28 fails because it requires additional undisclosed software. Claim 28 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "composite"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 29: Claim 29 is dependent on Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 29 fails because it requires additional undisclosed software. Claim 29 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "ensuring the integrity of said association . . ."). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 31: Claim 31 is dependent on Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 31 fails because it requires additional undisclosed software. Claim 31 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "codelivering"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 35: Claim 35 of the '891 patent fails the enablement requirement because

the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 35 (321:29-41), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 35. Claim 35 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure operating environment"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 35 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 36: Claim 36 of the '891 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 36 (321:44-57), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 36. Claim 36 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure operating environment system," "operatively connected," "logically associated with"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 36 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 39: Claim 39 is dependent on Claim 22 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 39 fails because it requires additional undisclosed software. Claim 39 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "persistently associating," "control arrangement"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 40: Claim 40 is dependent upon Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 40 fails because it requires additional undisclosed software. Claim 40 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "control arrangement"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 51: Claim 51 is dependent upon Claim 1 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 51 fails because it requires additional undisclosed software. Claim 51 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "end user electronic appliance," "secure processing step"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 53: Claim 53 is dependent upon Claim 22 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 53 fails because it requires additional undisclosed software. Claim 53 also fails the enablement requirement in light of the breadth of the subject matter

claimed (*e.g.* "end user electronic appliance"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 54: Claim 54 is dependent upon Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 54 fails because it requires additional undisclosed software. Claim 54 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "end user electronic appliance"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 56: Claim 56 is dependent upon Claim 35 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 56 fails because it requires additional undisclosed software. Claim 56 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "end user electronic appliance"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 57: Claim 57 is dependent upon Claim 36 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 57 fails because it requires additional undisclosed software. Claim 57 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "end user electronic appliance," "protected processing environment"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 58: Claim 58 is dependent upon Claim 1 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 58 fails because it requires additional undisclosed software. Claim 58 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "entity's control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 60: Claim 60 is dependent upon Claim 22 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 60 fails because it requires additional undisclosed software. Claim 60 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "supplying," "control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 61: Claim 61 is dependent upon Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 61 fails because it requires additional undisclosed software. Claim 61 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "providing"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 63: Claim 63 is dependent upon Claim 35 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 63 fails because it requires additional undisclosed software. Claim 63 also fails the enablement requirement in light of the breadth of the subject matter

claimed (*e.g.* "securely receiving"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 64: Claim 64 is dependent upon Claim 36 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 64 fails because it requires additional undisclosed software. Claim 64 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "controls"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 65: Claim 65 is dependent upon Claim 1 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 65 fails because it requires additional undisclosed software. Claim 65 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure processing environment"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 67: Claim 67 is dependent upon Claim 22 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 67 fails because it requires additional undisclosed software. Claim 67 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure processing environment"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 68: Claim 68 is dependent upon Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 68 fails because it requires additional undisclosed software. Claim 68 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure processing environment"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 70: Claim 70 is dependent upon Claim 35 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 70 fails because it requires additional undisclosed software. Claim 70 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure processing environment," "securely processing," "securely executing"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 71: Claim 71 is dependent upon Claim 1 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 71 fails because it requires additional undisclosed software. Claim 71 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "securely combining," "control arrangement"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 74: Claim 74 is dependent upon Claim 35 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 74 fails because it requires additional undisclosed software. Claim 74 also fails the enablement requirement in light of the breadth of the subject matter

claimed (*e.g.* "securely combining," "combined executable"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 75: Claim 75 is dependent upon Claim 36 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 75 fails because it requires additional undisclosed software. Claim 75 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "combined control arrangement"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 76: Claim 76 is dependent upon Claim 1 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 76 fails because it requires additional undisclosed software. Claim 76 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "securely receiving steps," "independently performed at different times"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 79: Claim 79 is dependent upon Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 79 fails because it requires additional undisclosed software.

Claim 81: Claim 81 is dependent upon Claim 35 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 81 fails because it requires additional undisclosed software. Claim 81 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "securely receiving steps"). The specification does not teach a person of ordinary

skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 82: Claim 82 is dependent upon Claim 36 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 82 fails because it requires additional undisclosed software. Claim 82 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "controls"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 84: Claim 84 is dependent upon Claim 1 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 84 fails because it requires additional undisclosed software. Claim 84 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "first/second entity's control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 86: Claim 86 is dependent upon Claim 26 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 86 fails because it requires additional undisclosed software. Claim 86 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 88: Claim 88 is dependent upon Claim 36 and thus fails the enablement

and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 88 fails because it requires additional undisclosed software. Claim 88 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 89: Claim 89 is dependent upon Claim 1 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 89 fails because it requires additional undisclosed software. Claim 89 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "control," "protected processing environment"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 91: Claim 91 is dependent upon Claim 22 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 91 fails because it requires additional undisclosed software. Claim 91 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 94: Claim 94 is dependent upon Claim 35 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 94 fails because it requires additional undisclosed software. Claim 94 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake

undue experimentation in order to make and use the invention across the full scope claimed.

Claim 95: Claim 95 is dependent upon Claim 36 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 95 fails because it requires additional undisclosed software. Claim 95 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

The '912 Patent

Claim 6: Claim 6 of the '912 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 6 (326:65-327:23), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 6. Claim 6 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "relatively lower level of security," "private portion characterized by . . . ," "accessing," "record"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 6 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 7: Claim 7 is dependent upon Claim 8 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 7 fails because it requires additional undisclosed software. Claim 7 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.*

"relatively higher/lower level of security"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 8: Claim 8 of the '912 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 8 (_____), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 8. Claim 8 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "higher/lower level of security," "execution space identifier," "assembling"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 8 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 9: Claim 9 is dependent upon Claim 8 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 9 fails because it requires additional undisclosed software.

Claim 13: Claim 13 is dependent upon Claim 8 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 13 fails because it requires additional undisclosed software. Claim 13 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "a security level higher that that of the execution space,"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the

invention across the full scope claimed.

Claim 14: Claim 14 is dependent upon Claim 13 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 14 fails because it requires additional undisclosed software.

Claim 35: Claim 35 of the '912 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 35 (330:27-57), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 35. Claim 35 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "second processing environment remote from first processing environment," "identification information"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 35 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

The '900 Patent

Claim 155: Claim 155 of the '900 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 155 (370:30-55), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 155. Claim 155 also fails the enablement requirement in light of

the breadth of the subject matter claimed (*e.g.* "host processing environment," "tamper resistant software designed to be loaded into said main memory . . .," "machine check programming which derives information . . .," "integrity programming"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 155 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 156: Claim 156 of the '900 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 156 (370:57-371:15), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 156. Claim 156 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "virtual distribution environment," "host processing environment," "tamper resistant software designed to be loaded into said main memory . . .," "machine check programming which derives information . . .," "integrity programming"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 156 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 157: Claim 157 of the '900 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 157 (371:16-42), both explicitly and

implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 157. Claim 157 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "virtual distribution environment," "host processing environment," "tamper resistant software designed to be loaded into said main memory . . .," "machine check programming which derives information . . .," "integrity programming"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 157 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

The '721 Patent

Claim 1: Claim 1 of the '721 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 1 (21:10-24), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 1. Claim 1 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "load module," "tamper resistance," "security level"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 1 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 5: Claim 5 of the '721 patent fails the enablement requirement because the

specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 5 (21:39-47), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 5. Claim 5 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "software verifying method," "specification"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 5 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 9: Claim 9 of the '721 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 9 (22:5-15), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 9. Claim 9 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "distinguishing between trusted and untrusted load modules . . ," "associated digital signature," "conditionally executing"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 9 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 14: Claim 14 of the '721 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 14 (22:44-51), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 14. Claim 14 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "arrangement within the first tamper resistant barrier that prevents . . .,"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 14 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 18: Claim 18 of the '721 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 18 (22:64-25:3), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 18. Claim 18 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "preventing the first computing arrangement . . ."). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 18 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 34: Claim 34 of the '721 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 34 (24:47-56), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 34. Claim 34 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure execution space," "security level"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 34 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 38: Claim 38 of the '721 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 38 (25:1-8), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 38. Claim 38 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "computing arrangement surrounded by a first tamper resistant barrier . . . ," "security level"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 38 fails the enablement and written description requirements of 35 U.S.C.

§ 112 ¶ 1.

The '019 Patent

Claim 1: Claim 1 of the '019 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 1 (319:46-320:7), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 1. Claim 1 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "associated control," "protected," transferring," "protected content file") The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 1 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 33: Claim 33 of the '019 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 33 (323:60-324:14), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 33. Claim 33 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means for incorporating," "means for transferring," "protected data") The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope

claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 33 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 34: Claim 34 is dependent upon Claim 33 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 34 fails because it requires additional undisclosed software. Claim 34 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means for applying"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 35: Claim 35 is dependent upon Claim 34 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 35 fails because it requires additional undisclosed software. Claim 35 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means for applying"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 41: Claim 41 of the '019 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 41 (325:7-29), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 41. Claim 41 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "virtual distribution environment") The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person

of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 41 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 42: Claim 42 is dependent upon Claim 41 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 42 fails because it requires additional undisclosed software. Claim 42 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "control," "protected information," "secure container"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 47: Claim 47 is dependent upon Claim 41 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 47 fails because it requires additional undisclosed software. Claim 47 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 52: Claim 52 is dependent upon Claim 41 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 52 fails because it requires additional undisclosed software. Claim 52 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "creating" "secure container," "site"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 53: Claim 53 is dependent upon Claim 52 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 53 fails because it requires additional undisclosed software. Claim 53 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "associated"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 54: Claim 54 is dependent upon Claim 53 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 54 fails because it requires additional undisclosed software. Claim 54 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "associated"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 55: Claim 55 is dependent upon Claim 54 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 55 fails because it requires additional undisclosed software. Claim 55 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "site"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 64: Claim 64 is dependent upon Claim 54 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 64 fails because it requires additional undisclosed software. Claim 64 also fails the enablement requirement in light of the breadth of the subject matter

claimed (*e.g.* "portion of said first protected information"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 76: Claim 76 is dependent upon Claim 41 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 76 fails because it requires additional undisclosed software. Claim 76 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure container," "contained"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 78: Claim 78 is dependent upon Claim 52 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 78 fails because it requires additional undisclosed software. Claim 78 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure container," "contained"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 81: Claim 81 of the '019 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 81 (328:9-23), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 81. Claim 81 also fails the enablement requirement in light of the breadth

of the subject matter claimed (*e.g.* "means for incorporating") The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 81 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 82: Claim 82 is dependent upon Claim 81 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 82 fails because it requires additional undisclosed software. Claim 82 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means for applying," "govern"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 83: Claim 83 is dependent upon Claim 82 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 83 fails because it requires additional undisclosed software. Claim 83 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "govern," "means for applying"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 85: Claim 85 of the '019 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 85 (328:28-56), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of

trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 85. Claim 85 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "creating," "copying," transferring"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 85 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 87: Claim 87 is dependent upon Claim 85 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 87 fails because it requires additional undisclosed software. Claim 87 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "copied," "protected information"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 89: Claim 89 is dependent upon Claim 85 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 89 fails because it requires additional undisclosed software. Claim 89 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "copying," "transferring"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 90: Claim 90 is dependent upon Claim 85 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 90 fails because it requires additional undisclosed software. Claim 90 also fails the enablement requirement in light of the breadth of the subject matter

claimed (*e.g.* "memory"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 93: Claim 93 is dependent upon Claim 85 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 93 fails because it requires additional undisclosed software. Claim 93 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "copying transferring"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 94: Claim 94 is dependent upon Claim 85 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 89 fails because it requires additional undisclosed software.

Claim 95: Claim 95 is dependent upon Claim 94 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 95 fails because it requires additional undisclosed software. Claim 95 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "copied," "protected information"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 96: Claim 96 of the '019 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 96 (329:38-330:12), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no

meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 96. Claim 96 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "virtual distribution environment," "protected information") The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 96 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

The '876 Patent

Claim 2: Claim 2 of the '876 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 2 (319:20-32), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 2. Claim 2 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means for . . . securely integrating," "value chain extended agreement"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 2 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 11: Claim 11 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 11 fails because it requires additional undisclosed software. Claim 11 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.*

"Virtual Distribution Environment"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 29: Claim 29 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 29 fails because it requires additional undisclosed software. Claim 29 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure control," "required terms"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 32: Claim 32 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 32 fails because it requires additional undisclosed software. Claim 32 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure control," "required terms"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 60: Claim 60 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 60 fails because it requires additional undisclosed software. Claim 60 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "secure control," "required terms"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 130: Claim 130 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 29 fails because it requires additional undisclosed software. Claim 29 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means for executing . . . control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 132: Claim 132 is dependent upon Claim 130 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 132 fails because it requires additional undisclosed software. Claim 132 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "protected processing environment"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 161: Claim 161 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 161 fails because it requires additional undisclosed software. Claim 161 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "machine executable controls"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 162: Claim 162 is dependent upon Claim 161 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 162 fails because it requires additional undisclosed software. Claim 162 also fails the enablement requirement in light of the breadth of the subject matter

claimed (*e.g.* "data descriptor data structures"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 170: Claim 170 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 170 fails because it requires additional undisclosed software. Claim 170 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means for creating a first secure control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 171: Claim 171 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 171 fails because it requires additional undisclosed software. Claim 171 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means for creating . . . secure control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 172: Claim 172 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 172 fails because it requires additional undisclosed software. Claim 172 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means . . . for securely integrating"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 329: Claim 329 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 329 fails because it requires additional undisclosed software. Claim 329 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means for creating . . . secure control"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 331: Claim 331 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 331 fails because it requires additional undisclosed software. Claim 331 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means . . . for securely integrating," "based on or compatible with . . ."). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 346: Claim 346 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 346 fails because it requires additional undisclosed software. Claim 346 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means by which said third control set governs . . ."). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 347: Claim 347 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 347 fails because it requires additional undisclosed software. Claim 347 also fails the enablement requirement in light of the breadth of the subject matter

claimed (*e.g.* "means by which said third control set governs the execution of at least one method"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 349: Claim 349 is dependent upon Claim 2 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 349 fails because it requires additional undisclosed software. Claim 349 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "means by which said third control set governs the execution of at least one procedure"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

The '181 Patent

Claim 48: Claim 48 of the '181 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 48 (48:17-38), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 48. Claim 48 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "narrowcasting selected digital information," secure node," "information derived in part from specified recipient's creation"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons

stated above with respect to all of the claims, Claim 48 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 59: Claim 59 is dependent upon Claim 48 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 59 fails because it requires additional undisclosed software. Claim 59 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 61: Claim 61 is dependent upon Claim 48 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 61 fails because it requires additional undisclosed software. Claim 61 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "entertainment information"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 63: Claim 63 is dependent upon Claim 48 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 63 fails because it requires additional undisclosed software. Claim 63 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "music information"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 67: Claim 67 is dependent upon Claim 48 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 67 fails because it requires additional undisclosed software.

Claim 67 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "digital certificate information"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 70: Claim 70 is dependent upon Claim 48 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 70 fails because it requires additional undisclosed software. Claim 70 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 72: Claim 72 is dependent upon Claim 48 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 72 fails because it requires additional undisclosed software. Claim 72 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 75: Claim 75 is dependent upon Claim 72 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 75 fails because it requires additional undisclosed software. Claim 75 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "acceptable clearinghouse," "rights and permissions clearinghouse"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 89: Claim 89 is dependent upon Claim 48 and thus fails the enablement

and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above.

Claim 91: Claim 91 of the '181 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 91 (86:47-87:4), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use the full scope of Claim 91. Claim 91 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "narrowcasting selected digital information," secure node," "information derived in part from specified recipient entity's creation"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 91 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

Claim 104: Claim 104 is dependent upon Claim 91 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 104 fails because it requires additional undisclosed software. Claim 104 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 109: Claim 109 is dependent upon Claim 91 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 109 fails because it requires additional undisclosed software. Claim 109 also fails the enablement requirement in light of the breadth of the subject matter claimed. The specification does not teach a person of ordinary skill in the art how to practice the

full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 114: Claim 114 is dependent upon Claim 91 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 114 fails because it requires additional undisclosed software. Claim 114 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "clearinghouse acceptable to rightsholders"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 117: Claim 117 is dependent upon Claim 114 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above. In addition, the limitation of Claim 117 fails because it requires additional undisclosed software. Claim 117 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "rights and permissions clearinghouse"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed.

Claim 131: Claim 131 is dependent upon Claim 91 and thus fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1 for the reasons stated above.

The '402 Patent

Claim 1: Claim 1 of the '402 patent fails the enablement requirement because the specification does not teach a person of ordinary skill in the relevant arts how to practice the purportedly disclosed invention without undue experimentation in the development of enabling software. Specifically, several limitations in Claim 1 (322:5-25), both explicitly and implicitly require software. Since no software is disclosed in the specification, and no meaningful programming guidance is provided, a person of skill in the art would have to engage a process of trial and error, perhaps followed by bottom up software development, in order to make and use

the full scope of Claim 1. Claim 1 also fails the enablement requirement in light of the breadth of the subject matter claimed (*e.g.* "creating," "having associated a first control" "value chain extended agreement," "transferring"). The specification does not teach a person of ordinary skill in the art how to practice the full scope of the claim, and a person of skill in the art would therefore be required to undertake undue experimentation in order to make and use the invention across the full scope claimed. For these reasons and for the reasons stated above with respect to all of the claims, Claim 1 fails the enablement and written description requirements of 35 U.S.C. § 112 ¶ 1.

IV. Patent L.R. 3-4

Each reference identified pursuant to PLR 3-3(a) but not in the prosecution history, and the documents referenced in PLR 3-4 that are sufficient to show the operation of the accused features of the products specifically and properly identified in InterTrust's PLR 3-1 Statements of September 2, 2003, has been or is being produced, or is otherwise available for inspection and copying. As set forth in greater detail in Microsoft's Motion to Strike InterTrust's Infringement Contentions (filed October 8, 2003), InterTrust's Infringement Contentions pursuant to PLR 3-1 largely fail to properly identify the "accused instrumentalities." Accordingly, Microsoft reserves its right to modify this production, if necessary. Microsoft has specifically sought, and has been granted, greater protection and confidentiality for its source code than that provided by Patent Local Rule 2-2. Source code for the Accused Instrumentalities is being made available for inspection at the offices of Orrick, Herrington & Sutcliffe LLP only in accordance with

///

///

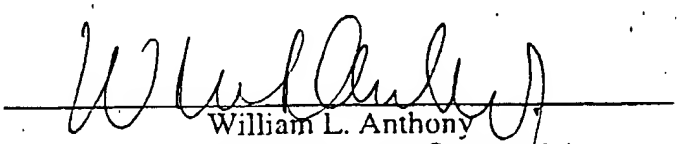
///

///

Magistrate James' Order of November 5, 2003. Microsoft does not concede that any source code made available for inspection (or any corresponding product or software) is or should be considered an Accused Instrumentality.

Dated: November 17, 2003

WILLIAM L. ANTHONY
ERIC L. WESENBERG
HEIDI L. KEEFE
ORRICK, HERRINGTON & SUTCLIFFE LLP



William L. Anthony
Attorneys for Defendant and Counterclaimant
MICROSOFT CORPORATION

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Lacy, Jack; Snyder, James; Maher, David; "Music on the Internet and the Intellectual Property Protection Problem"
Y	Yes	"The PowerTV White Paper", powertv.com website, Oct. 11, 1996
Y	Yes	Coutrot, Francois; Michon, Vincent; "A Single Conditional Access System for Satellite-Cable and Terrestrial TV" IEEE Transactions on Consumer Electronics, Vol. 35, No. 3, Aug. 1989
Y	Yes	"ISO 8583: Financial transaction card originated messages - Interchange message specifications", ISO, Dec. 15, 1993
Y	Yes	Harty, Kieran; Ho, Linda; "Case Study: The VISA Transaction Processing System", May 30, 1988
Y	Yes	U.S. 4,584,639; Apr. 22, 1986
	Yes	Denning, Dorothy E.; "Secure Personal Computing in an Insecure Network", Comm. of the ACM, Vol. 22, No. 8, Aug. 1979
	Yes	Muftic, Sead; "Security Mechnisms for Computer Networks", Computer Communications and Networking, 1989
Y	Yes	Kim, Gene H.; Spafford, Eugene H.; "The Design and Implementation of Tripwire: A File System Integrity Checker", COAST Laboratory, Purdue University, Nov. 19, 1993
Y	Yes	Choudhury, Abhijit K.; Maxemchuk, Nicholas F.; Paul, Sanjoy; Schulzrinne, Henning G.; "Copyright Protection for Electronic Publishing Over Computer Networks", IEEE Network, May/Jun., 1995
	Yes	Denning, Dorothy E.R.; Cryptorgraphy and Data Security, Addison-Wesley Publishing Company, 1982, Reprinted with corrections, Jan. 1983
	Yes	Hellman; "Multi-user Cryptographic Techniques"
	Yes	Diffie, Whitfield; Hellman, Martin E; "New Directions in Cryptography", Stanford University, 1976
Y	Yes	Kohl, J.; Neuman, C.; "The Kerberos Network Authentication Service (V5)", Network Working Group RFC 1510, Sep. 1993
	Yes	Diffie, Whitfield; van Oorschot, Paul C.; Weiner, Michael J.; "Authentication and Authenticated Key Exchanges", Sun Microsystems and Bell-Northern Research, Mar. 6, 1992
	Yes	Diffie, Whitfield; "The First Ten Years of Public-Key Cryptography", Proceedings of the IEEE, Vol. 76, No. 5, May, 1988
	Yes	Kohnfelder, Loren M.; "Towards a Practical Public-Key Cryptosystem", May, 1978
	Yes	Kaliski, Jr., Burton S.; "A Layman's Guide to a Subset of ASN.1, BER, and DER", RSA Laboratories Technical Note, 1991, Revised Nov. 1, 1993
Y	Yes	"PKCS #7: Cryptographic Message Syntax Standard", RSA Laboratories Technical Note, Ver. 1.5, Revised Nov. 1, 1993
	Yes	Walker, Stephen; "Notes from RSA Data Security Conference", Trusted Information Systems, Jan. 18, 1994
Y	Yes	Tygar, J.D.; Yee, Bennet; "Cryptography: It's Not Just for Electronic Mail Anymore", Carnegie Mellon University Tech. Report CMU-CS-93-107, Mar. 1, 1993
	Yes	U.S. 4,658,093; Apr. 14, 1987
Y	Yes	U.S. 4,405,829; Sep. 20, 1983
Y	Yes	Schneier, Bruce; Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, Inc., 1994

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
Y	Yes	Popek, Gerald J.; Kline, Charles S.; "Encryption and Secure Computer Networks", ACM Computing Surveys, Vol. 11, No. 4, Dec. 1979
	Yes	Diffie, Whitfield; Hellman, Martin E; "New Directions in Cryptography", Stanford University, 1976
	Yes	Castano, Silvana; Fugini, Mariagrazia; Martella, Giancarlo; Samarati, Pierangela; Database Security, ACM Press, 1994
Y	Yes	Thuraisingham, M.B.; "Mandatory Security in Object-Oriented Database Systems", OOPSLA '89 Proceedings, ACM, Oct. 1-6, 1989
Y	Yes	Olivier, Martin S.; von Solms, Sebastiaan H.; "A Taxomonmy for Secure Object-Oriented Databases", ACM Transactions on Database Systems, Vol. 19, No. 1, Mar. 1994
Y	Yes	Olivier, M.S.; von Solms, S.H.; "Building a Secure Database Using Self-Protecting Objects", Computers & Security, Vol. 11, No. 3, 1992
Y	Yes	Olivier, M.S.; von Solms, S.H.; "DISCO: A Discretionary Security Model for Object-oriented Databases", IT Security: The Need for International Cooperation, Elsevier Science Publishers B.V., 1992
Y	Yes	Oliver, Martin S.; "Secure Object-oriented Databases", Thesis for the degree of Doctor of Philosophy in Computer Science, Rand Afrikaans University, Dec. 1991
	Yes	R. Ahad, et al.; IRIS, 1992
Y	Yes	ORION l.k.a. ITASCA, MCC-Austin TX & Itasca Corp., 1985-1995
Y	Yes	Olivier, Martin S.; SECDB, 1990-1995
Y	Yes	"THOR: A Distributed Object-Oriented Database System", MIT
Y	Yes	Millen, Jonathan K.; Lunt, Teresa F.; "Security for Object-Oriented Database Systems", IEEE 0-8186-2825-1; 1992
	Yes	Choy, D.M. et al.; "A Digital Library System for Periodicals Distribution", May 1996
Y	Yes	Mathy, Laurent; "Features of The ACCOPI Multimedia Transport Service", Lecture Notes in Computer Science, No.1045, Proc. Of European Workshop IDMS'96, Mar. 1996; "Access Control and COpyright Protection for Images Security Technology for Graphics and Communication Systems - RACE M1005: ACCOPI", webpage, Security Projects at Fraunhofer-IGD, 2002; ACCOPI RACE Project M1005 Warning of ACCOPI web pages removal, UCL Laboratoire de telecommunications et teledetection
Y	Yes	"The Amide Products" web page; "Forum on Technology-Based Intellectual Property Management - Electronic Commerce for Content", IMA Intellectual Property Proceedings, Vol. 2, Jun. 1996
Y	Yes	Van Slype, Georges; "Natural Language Version of the generic CITED model -- Vol. I: Presentation of the generic model, ver. 3.0"; and "Vol. II: CITED usage monitoring system design for computer based applications, ver. 1.0", Project 5469, The CITED Consortium, Sep. 6, 1993

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
Y	Yes	"Technological Strategies for Protecting Intellectual Property in the Networked Multimedia Environment", IMA Intellectual Property Proceedings, Vol. 1, Issue 1, Jan. 1994
Y	Yes	COPICAT - 8195: "Copyright Ownership Projection in Computer-Assisted Training", ESPRIT, Dec. 1993; Kelman, Alistair; "Electronic Copyright Management: Possibilities and Problems", Scientists for Labor Presentation, Nov. 14, 1996
Y	Yes	Griswold, Gary N.; "A Method for Protecting Copyright on Networks", IMA Intellectual Property Proceedings, Vol. 1, Issue 1, Jan. 1994
	Yes	Erickson, John S.; "A Copyright Management System for Networked Interactive Multimedia", Proceedings of the 1995 Dartmouth Institute for Advanced Graduate Studies, 1995
	Yes	Burns, Christopher; "AAP Draft: Local Access and Usage Controls", Association of American Publishers Report, Apr. 13, 1995
Y	Yes	Choudhury, A.K.; Maxenchuk, N.F.; Paul, S.; Schulzrinne, H.G.; "Copyright Protection for Electronic Publishing over Computer Networks", Submitted to IEEE Network Magazine, Jun. 1994
	Yes	Wayner, Peter; Digital Copyright Protection, Academic Press, 1997
	Yes	"Cryptolope Containers Technology: A White Paper", IBM InfoMarket Business Development Group
	Yes	"Digital Rights Enforcement and Management: SuperDistribution of Cryptolopes", IBM
	Yes	Kaplan, Marc A.; "IBM Cryptolopes, SuperDistribution and Digital Rights Management", IBM, Dec. 30, 1996
	Yes	IP Workshop - CUPID: "Protocols and Services (ver. 1): An Architectural Overview", CNI, last update Nov. 20, 1997
Y	Yes	Patent Application EP 0 567 800 A1; Nov. 3, 1993
Y	Yes	Sibert, Olin; Bernstein, David; Van Wie, David; "The DigiBox: A Self-Protecting Container for Information Commerce", First USENIX Workshop on Electronic Commerce, Jul. 11-12, 1995
Y	Yes	Willett, Shawn; "Metered PCs: Is your system watching you?; Wave Systems beta tests new technology", IDG Communications, Inc. InfoWorld, May 2, 1994
Y	Yes	Weber, Robert; "Metering Technologies for Digital Intellectual Property - A Report to the International Federation of Reproduction Rights Organisations", International Federation of Reproduction on Rights Organisations, Northeast Consulting Resources, Inc., Oct. 1994
Y	Yes	TULIP Final Report, ISBN 0-444-82540-1, 1991, revised Sep. 18, 1996
	Yes	U.S. 5,634,012; May 27, 1997
	Yes	U.S. 5,715,403; Feb. 3, 1998
	Yes	U.S. 5,845,281; Dec. 1, 1998 (For Priority, Feb. 1, 1995)
Y	Yes	Brin, Sergey; Davis, James; Garcia-Molina, Hector; "Copy Detection Mechanism for Digital Documents", Stanford University
Y	Yes	Weber, Robert; "Digital Rights Management Technologies - A Report to the International Federation of Reproduction Rights Organisations", Northeast Consulting Resources, Inc., Oct. 1995

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Erickson, John S.; "Rights Management Through Enhanced Attribution", Presented at INET 96 Proceedings, Jun., 1996
	Yes	White, James E.; "Telescript: The Foundation for the Electronic Marketplace", Ver. 5.0, General Magic, Inc., Nov. 30, 1993
	Yes	Ketchpel, Steve P.; Garcia-Molina, Hector; Paepcke, Andreas; "Shopping Models: A Flexible Architecture for Information Commerce", Stanford University
	Yes	Lagoze, Carl; "A Secure Repository Design for Digital Libraries", D-Lib Magazine, Dec. 1995
Y	Yes	"Introduction to Smart Cards v. 1.0", Gemplus Card International, Mar. 21, 1991
	Yes	Abadi, M.; Burrows, M.; Kaufman, C.; Lampson, B.; "Authentication and Delegation with Smart-cards", Digital Equipment Corporation
Y	Yes	Tygar, J.D.; Yee, Bennet; "Dyad: A System for Using Physically Secure Coprocessors", IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1, Jan. 1994
	Yes	St. Johns, M.; "Draft Revised IP Security Option", Network Working Group, RFC 1038, Jan. 1988
	Yes	Galvin, J.; McCloghrie, K.; Davin, J.; "SNMP Security Protocols", Network Working Group RFC 1352, Jul., 1992
	Yes	U.S. 5,163,091; Nov. 10, 1992
	Yes	U.S. 5,355,474; Oct. 11 1994
Y	Yes	U.S. 5,678,170; Oct. 14, 1997
	Yes	U.S. 5,765,152; Jun. 9, 1998
	Yes	Shear, Victor; "Solutions for CD-ROM Pricing and Data Security Problems"
	Yes	Williams, Tony; "Microsoft Object Strategy", Microsoft PowerPoint presentation, 1990
Y	Yes	"OLE 2.0 Draft Content: Object Linking & Embedding", Microsoft, Jun. 5, 1991
	Yes	"Multimedia System Services Ver. 1.0", Hewlett-Packard, IBM, & SunSoft, 1993
	Yes	Draft "Request for Technology: Multimedia System Services", Ver. 1.1, Interactive Multimedia Association Compatibility Project, Oct. 16, 1992
	Yes	"Request for Technology: Multimedia System Services", Ver. 2.0, Interactive Multimedia Association Compatibility Project, Nov. 9, 1992
	Yes	Wobber, Edward; Abadi, Martin; Burrows, Mike; Lampson, Butler; "Authentication in the Taos Operating System", Digital Equipment Corporation, Dec. 10, 1993
	Yes	Custer, Helen; Inside Windows NT, Microsoft Press, pages 26-42 and 329-330, 1993
Y	Yes	Dynamic linking of SunOS
Y	Yes	Blaze, Matt, "A Cryptographic File System for Unix", preprint of paper to be presented at First ACM Conference on Communications and Computing Security, Nov. 3-5, 1993
	Yes	Gamble, Todd; "Implementing Execution Controls in Unix", USENIX Association, Proceedings of the Seventh Systems Administration Conference (LISA VII), Nov. 1-5, 1993

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
Y	Yes	Garfinkel, Simson; Spafford, Gene; Practical Unix Security, O'Reilly & Associates, Inc., 1994
	Yes	Blaze, Matt; Ioannidis, John; "The Architecture and Implementation of Network-Layer Security Under Unix", Columbia University and AT&T Bell Laboratories, 1994
	Yes	Sandhu, Ravi S.; "The Typed Access Matrix Model", Proceedings of IEEE Symposium on Security & Privacy, May 4-6, 1992
	Yes	Curry, David A.; Unix System Security: A Guide for Users and System Administrators, Addison-Wesley, 1992
Y	Yes	FreeBSD System manager's Manual "LDCONFIG", Oct. 3, 1993
	Yes	"Requirements for the Software License Management System", System Management Work Group, Rev. 3, Unix International, Jul. 23, 1992
Y	Yes	Film canister
Y	Yes	Safety deposit box
	Yes	Central Point Anti-Virus, Central Point Software, 1993
	Yes	Symantec Anti-Virus for Macintosh (a.k.a. SAM), Symantec, 1993
Y	Yes	VirusCheck and VirusScan, McAfee, 1993
	Yes	Goodman, Bill; Compactor Pro
	Yes	Enigma V.25
	Yes	StuffIt Deluxe v.1.5, v.3.0, v.3.5, Aladdin Systems, 1988-1994
Y	Yes	Harris, Jed; Ruben, Ira; "Bento Specification", Rev. 1.0d5, Apple Computer, Jul. 15, 1993
	Yes	Koenig, Andrew; "Automatic Software Distribution", USENIX Summer Conference Proceedings, Jun. 12-15, 1984
	Yes	Microsoft Internet Explorer v.2.0
	Yes	Think C: Object-Oriented Programming Manual, Symantec Corporation, 1989
	Yes	Think Pascal User Manual, Symantec Corporation, 1990
Y	Yes	Mori, Ryoichi; Kawahara, Masaji; "Superdistribution: The Concept and the Architecture", The Transactions of the IEICE, Vol. E 73, No. 7, Jul., 1990
	Yes	Epstein, Jeremy; Shugerman, Marvin; "A Trusted X Window System Server for Trusted Mach", USENIX Association, Mach Workshop, Aug. 30, 1990
	Yes	McCollum, Catherine J.; Messing, Judith R.; Notargiacomo, LouAnna; "Beyond the Pale of MAC and DAC -- Defining New Forms of Access Control", IEEE, 1990
	Yes	Abrams, Marshall D.; "Renewed Understanding of Access Control Policies", Proceedings of the 16th Computing National Security Conference, 1993
	Yes	Blaze, Matt; Feigenbaum, Joan; Lacy, Jack; "Decentralized Trust Management", Proc. IEEE Conference on Security and Privacy, May 1996
Y	Yes	Chaum, David; "Achieving Electronic Privacy", Scientific American, Aug. 1992
	Yes	UniverCD: The interactive, online library of product information from Cisco Systems, Cisco Systems, 1993
Y	Yes	DCE
	Yes	Fine, Todd; Minear, Spencer E.; "Assuring Distributed Trusted Mach", Secure Computing Corporation
	Yes	U.S. 5,412,717; May 2, 1995

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Fugini, M.G.; Zicari, R.; "Authorization and Access Control in the Office-Net System", Computer Security in the Age of Information, IFIP, 1989
	Yes	Abadi, M.; Burros, M.; Lampson, B.; Plotkin, G.; "A Calculus for Access Control in Distributed Systems", Digital Equipment Corporation, Feb. 28, 1991, revised Aug. 28, 1991
Y	Yes	Lampson, Butler; Abadi, Martin; Burrows, Michael; Wobber, Edward; "Authentication in Distributed Systems: Theory and Practice", Digital Equipment Corporation, 1992
	Yes	Rivest, Ronald L.; Lampson, Butler; "SDSI - A Simple Distributed Security Infrastructure", MIT and Microsoft Corporation, Apr. 30, 1996
	Yes	Thompson, Victoria P.; Wentz, F. Stan; "A Concept for Certification of an Army MLS Management Information System", Proceedings of the 16th National Computer Security Conference, Sep. 20-23, 1993
	Yes	Frederick, Keith P.; "Certification and Accreditation Approach", Air Force Cryptologic Support Center (OL-FP)
Y	Yes	PCT Application WO 96/27155; Published Sep. 6, 1996
	Yes	U.S. 5,910,987; Jun. 8, 1999
Y	Yes	Rozenblit, Moshe; "Secure Software Distribution", IEEE 0-7803-1811-0/94, 1994
Y	Yes	Stefik, Mark; Internet Dreams: Archetypes, Myths, and Metaphors, "Letting Loose the Light: Igniting Commerce in Electronic Publication", The MIT Press, 1996
	Yes	AT&T PersonaLink, [Before Feb. 13, 1995]
	Yes	Neuman, B. Clifford; "Proxy-Based Authorization and Accounting for Distributed Systems", Proceedings of the 13th Int'l Conference on Distributed Computing Systems, May 1993
Y	Yes	Tygar, J.D.; Yee, Bennet S.; (R. Rashid, ed.); "Strongbox: A System for Self-Securing Programs"
	Yes	Yee, Bennet; Tygar, J.D.; "Secure Coprocessors in Electronic Commerce Applications", Proceedings of the First USENIX Workshop on Electronic Commerce, Jul. 1995
	Yes	U.S. 4,278,837; Jul. 14, 1981
	Yes	U.S. 3,806,874; Apr. 23, 1974
Y	Yes	U.S. 4,748,561; May 31, 1988
Y	Yes	U.S. 4,796,220; Jan. 3, 1989
	Yes	U.S. 4,817,140; Mar. 28, 1989
Y	Yes	U.S. 4,866,769; Sep. 12, 1989
Y	Yes	U.S. 5,014,234; May 7, 1991
Y	Yes	U.S. 5,113,518; May 12, 1992
	Yes	U.S. 5,204,897; Apr. 20, 1993
	Yes	U.S. 5,218,605; Jun. 8, 1993
Y	Yes	U.S. 5,260,999; Nov. 9, 1993
Y	Yes	U.S. 5,291,598; Mar. 1, 1994
Y	Yes	U.S. 5,337,357; Aug. 9, 1994
	Yes	U.S. 5,421,006; May 30, 1995
	Yes	U.S. 5,438,508; Aug. 1, 1995
	Yes	U.S. 5,490,216; Feb. 6, 1996
Y	Yes	U.S. 5,603,031; Feb. 11, 1997

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	U.S. 5,692,047; Nov. 25, 1997
	Yes	U.S. 5,724,425; Mar. 3, 1998
	Yes	U.S. 5,940,504; Aug. 17, 1999
	Yes	U.S. 5,978,484; Nov. 2, 1999
	Yes	U.S. 6,016,393; Jan. 18, 2000
	Yes	Woo, Thomas Y.C.; Lam, Siman S.; "A Framework for Distributed Authorization", 1st Conf. Computer & Comm. Security, ACM, Nov., 1993
	Yes	Sandhu, Ravi S.; Suri, Gurpreet S.; "Implementation Considerations for the Typed Access Matrix Model in a Distributed Environment", Proc. Of the 15th National Computer Security Conference, Oct. 1992
	Yes	O'Conner, MaryAnn; "New Distribution Options for Electonic Publishers: iOpener Data Encryption and Metering System for CD-ROM Use", CD-ROM Professional, Vol 7, No. 2, ISSN 1409-0833, Mar. 1994
	Yes	Herzberg, A; Karmi, G; "On Software Protection", Proceedings of the 4th Jerusalem Conference on Information Technology (JCIT), IEE Computer Society Press, Apr. 1984
	Yes	Smith, Mary Grace; Weber, Robert; "A New Set of Rules for Information Commerce: Rights-Protection Technologies and Personalized-Information Comerce Will Affect All Knowledge Workers", CommunicationsWeek, Nov. 6, 1995
Y	Yes	DOD "Rainbow Series"
	Yes	Rosenthal, Doug; "EINet: A Secure, Open Network for Electronic Commerce", IEEE, 1994
Y	Yes	Patent Application EP 0 367 700 A2; May 9, 1990
Y	Yes	Hauser, R.; Bauknecht, K.; "LTTP Protection – A Pragmatic Approach to Licensing", Institut fur Informatik, Universitat Zurich, Jan. 13, 1994
	Yes	"Multimedia Mixed Object Envelopes Supporting a Graduated Fee Scheme via Encryption"; IBM Technical Disclosure Bulletin, Vol. 37, No. 3, Mar. 1994
	Yes	Cox, Brad; "No Silver Bullet Revisted", American Programmer Journal, Nov. 1995
	Yes	"Privacy and the NII: Safeguarding Telecommunications-Related Personal Information", U.S. Dept. of Commerce, Oct. 1995
	Yes	Joseph Ebersole, Protecting Intellectual Property Rights on the Information Superhighways, Mar. 1994
Y	Yes	Herzberg, Amir; Printer, Shlomit S.; "Public Protection of Software", ACM Transactions on Computer Systems, Vol. 5, No. 4, Nov. 1987
	Yes	Hickman, Kipp E.B.; SSL 2.0 Protocol Specification
	Yes	Gosler, James; "Software Protection: Myth or Reality", Lecture Notes in Computer Science, Advances in Cryptology - Crypto '85 Proceedings, 1985
	Yes	Aucsmith, David; "Tamper Resistent Software: An Implementation", IAL
	Yes	U.S. Patent No. 5,671,279; Sept. 23, 1997
Y	Yes	Kahn, Robert; Wilensky, Robert; "A Framework for Distributed Digital Object Services", Corporation for National Research Initiatives, May 13, 1995
Y	Yes	Gasser, Morrie; Goldstein, Andy; Kaufman, Charlie; Lampson, B; "The Digital Distributed System Security Architecture", Proceedings of 1989 National Computer Security Conference, 1989

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
Y	Yes	Neuman, B. Clifford; Ts'o, Theodore; "Kerberos: An Authentication Service for Computer Networks", IEEE Communications Magazine, Sep. 1994
	Yes	Reiher, Peter; Page, Jr., Thomas; Popek, Gerald; Cook, Jeff; Crocker, Stephen; "Truffles -- Secure File Sharing With Minimal System Administrator Intervention", UCLA, Trusted Information Systems
Y	Yes	Reiher, Peter; Page, Jr., Thomas; Popek, Gerald; Cook, Jeff; Crocker, Stephen; "Truffles -- A Secure Service for Widespread File Sharing", UCLA, Trusted Information Systems
Y	Yes	"ISO, Open Systems Interconnection: Security Architecture, ISO 7498/1", 1988
Y	Yes	"ISO, Open Systems Interconnection: Security Architecture, ISO 7498/2", ISO, 1988
	Yes	U.S. 5,222,134; Jun. 22, 1993
	Yes	Rindfrey, Jochen; "Security in the World Wide Web", Fraunhofer Institute for Computer Graphics, Dec. 1996
	Yes	Finin, Tim; Fritzson, Rich; McKay, Don; "A Language and Protocol to Support Intelligent Agent Interoperability", Proceedings of the CE & CALS Washington '92 Conference, Apr. 1992
Y	Yes	Winslet, Marianne; Smith, Kenneth; Qian, Xiaolei; "Formal Query Languages for Secure Relational Databases", ACM Transactions on Database Systems, Vol. 19, No. 4, Dec. 1994
	Yes	Jones, V.E.; Ching, N.; Winslett, M.; "Credentials for Privacy and Interoperation", University of Illinois at Urbana-Champaign
	Yes	Greenwald, Steven J.; Newman-Wolfe, Richard E.; "The Distributed Compartment Model for Resource Management and Access Control", Technical Report Number TR94-035, The University of Florida, Oct. 1994
Y	Yes	Moffett, Jonathan D.; "Delegation of Authority Using Domain-Based Access Rules", thesis, Imperial College of Science, Technology & Medicine, University of London, Jul., 1990
Y	Yes	Lagoze, Carl; McGrath, Robert; Overly, Ed; Yeager, Nancy; "A Design for Inter-Operable Secure Object Stores (ISOS)", Cornell University, NCSA, CNRI, Nov. 7, 1995
	Yes	Aharonian, Gregory; "Software Patents - Relative Comparison of EPO/PTO/JPO Software Searching Capabilities", Source Translation & Optimization
	Yes	Gaster, Jens L.; "Authors' Rights and Neighbouring Rights in the Information Society", DG XV/E/4, European Commission
	Yes	"Europe and The Global Information Society Recommendations to the European Council", Bangemann Report, www.medicif.org web pages, Global Information Society, May, 26, 1994
	Yes	Bernstein, David; Lenowitz, Erwin; "Copyrights, Distribution Chains, Integrity, and Privacy: The Need for a Standards-Based Solution", Electronic Publishing Resources
	Yes	Rubin, A.D.; Honeyman, P.; "Long Running Jobs in an Authenticated Environment", CITI Technical Report 93-1, Center for Information Technology Integration, Mar. 29, 1993
	Yes	Sammer, Peter; Ausserhofer, Andreas; "New Tools for the Internet", Joanneum Research, Graz University of Technology

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Eizenberg, Gerard, "Contribution of Information Technology Security to Intellectual Property Protection", CERT-DERI
	Yes	Antonelli, C.J.; Doster, W.A.; Honeyman, P.; "Access Control in a Workstation-Based Distributed Computing Environment", CITI Technical Report 90-2, Jul. 17, 1990
	Yes	Lord, S.P.; Pope, N.H.; Stepney, Susan; "Access Management in Multi-Administration Networks", IEE 2nd International Conference on Secure Communication Systems, 1986
	Yes	Stepney, Susan; Lord, Stephen P.; "Formal Specification of an Access Control System", Software-Practice and Experience, Vol 17(9), 1987
	Yes	Brunnstein, Klaus; Sint, Peter P.; "Intellectual Property Rights and New Technologies", Proceedings of the KnowRight'95 Conference, Aug. 1995
	Yes	Rubin, A.D.; Honeyman, P.; "Formal Methods for the Analysis of Authentication Protocols CITI Technical Report 93-7", Center for Information Technology Integration, Nov. 8, 1993
	Yes	Lexis/WestLaw
Y	Yes	U.S. 6,135,646; Oct. 24, 2000
	Yes	Bishop, Matt; "Privacy-Enhanced Electronic Mail", Privacy and Security Researh Group, IAB
Y	Yes	Kim, Won; Ballou, Nat; Chou, Hong-Tai; Garza, Jorge F.; Woelk, Darrell; "Features of the ORION Object-Oriented Database System"
	Yes	"Key Management Using ANSI X9.17", Federal Information Processing Standards Publication 171, U.S. Department of Commerce, Apr. 27, 1992
	Yes	"S/PAY: RSA's Developer's Suite for Secure Electronic Transactions (SET)", RSA Data Security, Inc., 1997
	Yes	Perlman, Bill; "A Working Anti-Taping System for Cable Pay-Per-View", IEEE Trans. On Consumer Electronics, Vol. 35, No.3, Aug. 1989
Y	Yes	Organick, Elliott I.; "The Multics System: An Examination of Its Structure", MIT Press, 1972
Y	Yes	Cina Jr., Vincent J.; White, Star R.; Comerford, Liam; "ABYSS: A Basic Yorktown Security System PC Software Asset Protection Concepts", IBM Research Report Number RC 12401, IBM Thomas J. Watson Research Center, Dec. 18, 1986
Y	Yes	White, Steve R.; Comerford, Liam; "ABYSS: An Architecture for Software Protection", IEEE Transactions on Software Engineering, Vol. 16, No. 6, Jun. 1990
Y	Yes	Davies, D.W.; Price, W.L.; Security for Computer Networks, John Wiley & Sons, 1984
	Yes	"MSDN - INF: LAN Manager 2,1 Server Autotuning (Part 2)", PSS ID Number Q80078, Microsoft, Feb. 1993
Y	Yes	"MSDN - License Service Application Programming Interface", API Specification v1.02, Microsoft, Jan. 1993

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	"International Infrastructure Standards Panel: IISP Need #31 - Containers or Secure Packaging; IISP Need #32 - Authentication of Content; IISP Need #33 - Control Enforcement; IISP Need #34 - Billing and Payment; IISP Need #35 - Reporting" ANSI Online, Sep. 18, 1995
Y	Yes	"Cryptographic API Specification", Version 0.6, Microsoft, Mar. 1995
	Yes	Everett, David B.; "Smart Card Tutorial - Part 1", Sep. 1992
Y	Yes	Paradinas, Pierre; Vandewalle, Jean-Jacques; "New Directions for Integrated Circuit Cards Operating Systems"
Y	Yes	Hauser, Ralf; "Control of Information Distribution and Access", Dissertation Der Wirtschaftswissenschaftlichen Fakultat Der Universitat Zurich, May 31, 1995
	Yes	Rindfrey, Jochen; "Towards an Equitable System for Access Control and Copyright Protection in Broadcast Image Services: The Equicrypt Approach", Fraunhofer Institute for Computer Graphics
	Yes	Wells, Rob; Odyssey of Plastic Purchase: 20-Second Round-Trip, Associated Press, Dec. 1993
	Yes	Payment Systems: Strategic Choices for the Future, Hitachi Research Institute; Institute of Advanced Business Systems, Hitachi, Ltd., 1993
	Yes	"EFT Network Data Book - 1993 Edition", Bank Network News, Vol. 11, No. 13, Nov. 1992
	Yes	"American National Standard: Specification for Financial Message Exchange Between Card Acceptor and Acquirer, X9.15", American Banker's Association, 1990
	Yes	"ISO 7813-1987 Identification Cards - Financial Transaction Cards", ISO, 1987
Y	Yes	MSDN Issue: Summer 1992; Vol. No.: 0 (Beta); 1 Disk, Microsoft, 1992
Y	Yes	MSDN Issue: Sep. 1992; Vol. No.: 1; 1 Disk, Microsoft, Sep. 1992
Y	Yes	MSDN Issue: Jan 1993; Vol. No. 2; 1 Disk, Microsoft, Jan. 1993
Y	Yes	MSDN Issue: Apr. 1993; Vol. No. 3; 1 Disk, Microsoft, Apr. 1993
Y	Yes	MSDN Issue: Summer 1993; Vol. No. 4; 1 Disk, Microsoft, Jul. 1993
Y	Yes	MSDN Issue: Fall 1993; Vol. No. 5; 1 Disk, Microsoft, Oct. 1993
Y	Yes	MSDN Issue: Winter 1994; Vol. No. 6; 1 Disk, Microsoft, Jan. 1994
Y	Yes	MSDN Issue: Apr. 1994; Vol. No. 7; 1 Disk, Microsoft, Apr. 1994
Y	Yes	MSDN Issue: Jul. 1994; Vol. 8; 1 Disk, Microsoft, Jul. 1994
Y	Yes	MSDN Issue: Oct. 1994; Vol. 9; 1 Disk, Microsoft, Oct. 1994
Y	Yes	MSDN Issue: Jan 1995; Vol. 10; 1 Disk, Microsoft, Jan. 1995
Y	Yes	MSDN Issue: Apr. 1995; Vol. 11; 1 Disk, Microsoft, Apr. 1995
Y	Yes	MSDN Issue: Jul. 1995; Vol. 12; 1 Disk, Microsoft, Jul. 1995
Y	Yes	MSDN Issue: Oct. 1995; Vol. 13; 1 Disk, Microsoft, Oct. 1995
Y	Yes	MSDN Issue: Jan 1996; Vol. 14; 2 Disks, Microsoft, Jan. 1996
Y	Yes	MSDN Issue: Apr. 1996; Vol. 15; 2 Disks, Microsoft, Apr. 1996
Y	Yes	MSDN Issue: Jul. 1996; Vol. 16; 1 Disk, Microsoft, Jul. 1996
Y	Yes	MSDN Issue: Oct. 1996; Vol. 17; 2 Disks, Microsoft, Oct. 1996
Y	Yes	MSDN Issue: Jan 1997; Vol. 18; 2 Disks, Microsoft, Jan. 1997
Y	Yes	MSDN Issue: 16-Bit Archive 1997; Vol. NA; 1 Disk, Microsoft, Jan. 1997

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
Y	Yes	MSDN Issue: Apr. 1997; Vol. No. 20; 2 Disks, Microsoft, Apr. 1997
Y	Yes	MSDN Issue: Jul. 1997; Vol. No. 21; 2 Disks, Microsoft, Jul. 1997
Y	Yes	MSDN Issue: Oct. 1997; Vol. No. 24; 2 Disks, Microsoft, Oct. 1997
Y	Yes	MSDN Issue: Visual Studio 1997; Vol. No. 191; 1 Disk, Microsoft, 1997
Y	Yes	MSDN Issue: Jan. 1998; Vol. No. 27; 2 Disks, Microsoft, Jan. 1998
Y	Yes	MSDN Issue: Apr. 1998; Vol. No. 30; 2 Disks, Microsoft, Apr. 1998
Y	Yes	MSDN Issue: Jul. 1998; Vol. No. 33; 3 Disks, Microsoft, Jul. 1998
Y	Yes	MSDN Issue: Oct. 1998; Vol. No.: None; 3 Disks, Microsoft, Oct. 1998
Y	Yes	MSDN Issue: Jan 1999; Vol. No.: None; 3 Disks, Microsoft, Jan. 1999
Y	Yes	MSDN Issue: Apr. 1999; Vol. No.: None; 3 Disks, Microsoft, Apr. 1999
Y	Yes	MSDN Issue: Jul. 1999; Vol. No.: None; 3 Disks, Microsoft, Jul. 1999
Y	Yes	MSDN Issue: Oct. 1999; Vol. No.: None; 3 Disks, Microsoft, Oct. 1999
Y	Yes	Chaum, David; Smart Card 2000, Selected Papers from the Second International Smart Card 2000 Conference, Oct. 4-6, 1989
Y	Yes	CD Jukebox
	Yes	U.S. Patent No. 4,926,480; May 15, 1990
	Yes	U.S. Patent No. 4,529,870; Jul. 16, 1985
	Yes	Meyer, Carl H.; Matyas, Stephen M.; Cryptography: A New Dimension in Computer Security, John Wiley & Sons, New York, 1982
	Yes	"Interchange Message Specification for Debit and Credit Card Message Exchange Among Financial Institutions", American National Standard, Accredited Standards Committee X9-Financial Services Committee, ANSI X9.2-1988, American Bankers Association, May 16, 1988
Y	Yes	Excerpts from Jul. 1993 MSDN disks, Jul. 1993
Y	Yes	Cox, Benjamin; Tygar, J.D.; Sirbu, Marvin; "NetBill Security and Transaction Protocol", Carnegie Mellon University
	Yes	Cox, Brad; "What if there is a Silver Bullet and the competition gets it first?", Journal of Object-oriented Programming, Jun. 1992
Y	Yes	"CITED Final Report: A Guide to CITED Documentation", ESPRIT, Project 5469, ISBN 0-7123-2115-2, The CITED Consortium, Sep. 1994
Y	Yes	Boisson, Jean-Francois; "1 - Business Perspectives and Requirements, 2 - The CITED Project: keys and knowledge", CITED 5469
Y	Yes	Van Slype, Georges; "Knowledge Economy: future trends", CITED 5469
Y	Yes	Boisson, Jean-Francois; "Software components: deliverable Trial Offer", CITED 5469
Y	Yes	Van Slype, Georges; "The CITED approach, Ver. 4.0", ESPRIT II, Project 5469, The CITED Consortium, Apr. 20, 1994
Y	Yes	Moens, Jan; "Report on the users requirements, Ver. 1.0", ESPRIT II, Project 5469, The CITED Consortium, Nov. 27, 1991
Y	Yes	Schulze, Dr. J.; "Case of application of the generic CITED Model to the CITEDisation in the software distribution process", ESPRIT II, Project , Jan. 12, 1993
Y	Yes	Moens, Jan; "Case of application of the generic CITED Model to the CITEDisation of a directory database on CD-ROM, Ver. 2.0", ESPRIT II, Project 5469, The CITED Consortium, Nov. 30, 1992
Y	Yes	Pijnenborg, Mari F.J.; "CITED Final Report", Elsevier Science B.V., Apr. 1994

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
Y	Yes	Boisson, Jean-Francois; "How to CITEDise application: Guidelines and examples", CITED 5469
Y	Yes	Nguyen, Thanh; Saint Etienne, Patricia Louise (SAGEM); "Guidelines for Validation of a CITED System", CITED 5469, SA-21-40-003, Jul. 4, 1994
Y	Yes	Van Slype, Georges; "The future of CITED: a feasibility study, Ver. 1.1 - Vol. I: Summary report and recommendations", ESPRIT II, Project 5469, The CITED Consortium, Mar. 28, 1994
Y	Yes	Van Slype, Georges; "The future of CITED: a feasibility study, Ver. 1.1 - Vol. III: Draft CITED interchange formats", ESPRIT II, Project 5469, The CITED Consortium, Mar. 28, 1994
Y	Yes	"CITED: Copyright in Transmitted Electronic Documents, Special Interest Group", CITED, Meeting, Heathrow, Sep. 22, 1993
Y	Yes	Miscellaneous letters from Georges Van Slype at Bureau Van Dijk, Mar. 30, 1995
Y	Yes	Pijnenborg, Mari F.J.; "auteursrecht en de digitale bibliotheek", 195 Open, Jan. 37, 1995
Y	Yes	Miscellaneous letters from Georges Van Slype at Bureau Van Dijk, Feb. 13, 1995, Nov. 2, 1994
Y	Yes	Van Slype, Georges; "PL4 RACE/ACCOPI Workshop on Conditional Access and Copyright Protection", ESPRIT II, Project 5469, The CITED Consortium, Nov. 9, 1994
Y	Yes	Miscellaneous letters from G. Van Slype at Bureau Van Dijk, Sep. 12, 1994, Sep. 1994, May 11, 1994, May 10, 1994, May 6, 1994, May 4, 1994, Apr. 21, 1994, Apr. 20, 1994
Y	Yes	Letter re: ESPRIT III – Project 5469 (CITED) from A. Stajano at Commission of the European Communities, Oct. 7, 1993
Y	Yes	ESPRIT Project: 5469: Contract Amendment Number: 2; Commission of the European Communities, Sep. 16, 1993
Y	Yes	Miscellaneous letters from George Van Slype at Bureau Van Dijk, Apr. 19, 1994, Apr. 18, 1994, Apr. 11, 1994, Apr. 6, 1994
Y	Yes	"The Future of Cited: A Feasibility Study", ESPRIT II, Project 5469, The CITED Consortium Apr. 15, 1994
Y	Yes	Miscellaneous letters from Bureau Van Dijk, Mar. 30, 1994, Mar. 24, 1994, Feb. 10, 1994, Feb. 10, 1994
Y	Yes	Handwritten note re: GVS and AJL, Mar. 2, 1994
Y	Yes	Miscellaneous letters from Bureau Van Dijk, Feb. 9, 1994, Jan. 27, 1994, Jan. 19, 1994, Jan. 12, 1994, Dec. 22, 1993, Nov. 30, 1993, Nov. 22, 1993, Dec. 6, 1993, Nov. 16, 1993, Oct. 15, 1993, Oct. 7, 1993, Oct. 4, 1993, Sep. 20, 1993, Sep. 7, 1993, May 19, 1993, Oct. 13, 1993
Y	Yes	Bureau Van Dijk Management Report for Task 4.5: Feasibility Study of the Cited Agency, 1992-1993
Y	Yes	Bureau van Dijk: Gestion des contrats; 497C C.C.E. : CITED (SUITE), Feb. 1993
Y	Yes	"CITED: Preparation of the CITED model functional requirements specifications – Discussion paper (revision 1)", Bureau Van Dijk, Jan. 16, 1991

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
Y	Yes	"CITED: Preparation of the CITED Model Functional Requirements Specifications – Report of the interview with OXFORD UNIVERSITE PRESS, CITED part", Bureau Van Dijk, Feb. 27, 1991
Y	Yes	"CITED: Preparation of the CITED Model Functional Requirements Specifications – Reports of the interviews with five CITED Partners" (Partners: Sagem, Telesystemes, NTE, Elsevier, Oxford University Press), Bureau Van Dijk, Apr. 5, 1991
Y	Yes	"CITED: Preparation of the CITED Model Functional Requirements Specifications – Reports of the interviews with Seven International Organizations: EBU, ECMA, ELDA, IFPI, IFTC, STM, WIPO", Bureau Van Dijk, May 27, 1991
Y	Yes	Van Slype, Georges; Moens, Jan; Vannieuwenhuyse, Lawrence; "The future of CITED: a feasibility study", ESPRIT II, Project 5469, The CITED Consortium, Nov. 15, 1993
Y	Yes	Van Slype, Georges; "Draft CITED interchange formats, Ver. 1.0", ESPRIT II, Project 5469, The CITED Consortium, Jan. 28, 1994
Y	Yes	Miscellaneous letter from Georges Van Slype at Bureau Van Dijk, Feb. 28, 1994
Y	Yes	Van Slype, Georges; "The future of CITED : a feasibility study, Ver. 1.0 – Vol. I: Summary report and recommendations", ESPRIT II, Project 5469, The CITED Consortium, Feb. 28, 1994
Y	Yes	Van Slype, Georges; Moens, Jan; Vannieuwenhuyse, Laurence; "The future of CITED: a feasibility study, Ver. 1.0 – Vol. II: Full report", ESPRIT II, Project 5469, The CITED Consortium, Feb. 28, 1994
Y	Yes	Van Slype, Georges "The future of CITED: a feasibility study, Ver. 1.1 – Vol. III: Draft CITED interchange formats", ESPRIT II, Project 5469, The CITED Consortium, Feb. 28, 1994
Y	Yes	"The Future of Cited: A Feasibility Study", ESPRIT II, Project 5469, CITED Project Review, Apr. 15, 1994
Y	Yes	Van Slype, Georges; "PL4 RACE/ACCOPI Workshop on Conditional Access and Copyright Protection", ESPRIT II, Project 5469, Presentation of the CITED, Nov. 9, 1994
Y	Yes	Van Slype, Georges; "Natural Language version of the generic CITED model, Ver. 4.2 – Vol. I: Presentation of the generic model", ESPRIT II, Project 5469, The CITED Consortium, May 8, 1995
Y	Yes	Van Slype, Georges; "Natural language version of the generic CITED model, Ver. 2.1 – Vol. II ECMS (Electronic Copyright Management System) design for computer based applications", ESPRIT II, Project 5469, The CITED Consortium, May 8, 1995
	Yes	Cousins, Steve B.; Ketchpel, Steven P.; Paepcke, Andreas; Garcia-Molina, Hector; Hassan, Scott W.; Roscheisen, Martin; "InterPay: Managing Multiple Payment Mechanisms in Digital Libraries"
	Yes	"PKCS #5: Password-Based Encryption Standard", An RSA Laboratories Technical Note, Ver. 1.5, 1991-1993, Revised Nov. 1, 1993
	Yes	"PKCS #8: Private-Key Information Syntax Standard", An RSA Laboratories Technical Note, Ver. 1.2, 1991-1993, Revised Nov. 1, 1993
	Yes	"PKCS #10: Certification Request Syntax Standard", An RSA Laboratories Technical Note, Ver. 1.0, Nov. 1, 1993

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	"PKCS #11: Cryptographic Token Interface Standard", An RSA Laboratories Technical Note, Ver. 2.0, Apr. 15, 1997
	Yes	"PKCS 12 v1.0: Personal Information Exchange Syntax", RSA Laboratories, Jun. 24, 1999
	Yes	"PKCS #13: Elliptic Curve Cryptography Standard", RSA Security, Jan. 12, 1998
	Yes	"PKCS #15 v1.0: Cryptographic Token Information Format Standard", RSA Laboratories, Apr. 23, 1999
	Yes	U.S. 5,335,346; Aug. 2, 1994
Y	Yes	Garfinkel, Simson; Spafford, Gene; Practical UNIX Security, O'Reilly & Associates, Inc., 1991
Y	Yes	Merkle, Ralph C., "Protocols for Public Key Cryptosystems", IEEE, 1980
	Yes	Kaner, Cem; Falk, Jack; Nguyen, Hung Quoc; Testing Computer Software, Second Edition, Van Nostrand Reinhold, 1988
	Yes	Press, Jim; Bunting, Angela "A New Approach to Cryptographic Facility Design", ICL Mid-Range Systems Division Reading, Berks, UK
Y		US 6,256,668; Jul. 3, 2001
Y		Kim, Gene H.; Spafford, Eugene H.; "Experiences with Tripwire: Using Integrity Checkers for Intrusion Detection", Purdue Technical Report CSD-TR-94-012, Feb. 21, 1994
Y		"Technical Description: Pay-Per-View Copy Protection", Macrovision, Jun. 1994
Y		Reali, Patti; "Copy Protection: The answer to pay per view's prayers?", TVRO Dealer, Dec. 1994
		Swedlow, Tracy; "2000: Interactive Enhanced Television: A Historical and Critical Perspective", Interactive TV Today
		Various articles from EE Times, Week of Oct. 2, 1995
		"Digital Broadband Delivery System, Phase 1.0, System Overview", Revision 1.0, Scientific Atlanta, 1997
		Langelaar, G.C. "Overview of protection methods in existing TV and storage devices", SMS-TUD-609-1, Final Ver. 1.2, Feb. 26, 1996
Y		Solomon, A.; "PC Viruses: Detection, Analysis, and Cure", Springer Verlag 1991.
Y		Galaxy, Opcode Systems, 1991-1994
Y		Unix System V & BSD & GNU versions prior to Feb 22, 1996
Y		US 5,673,316; Sep. 30, 1997
Y		17 USCA sections 1001 - 1010, Chapter 10 Digital Audio Recording Devices and Media, 1996
		Hill, Will; Hollan, Jim; "History-Enriched Digital Objects", Computer Graphics and Interactive Media Research Group; Bell Communications Research, 1993
		Hill, William; Hollan, James D.; "Edit Wear and Read Wear", Computer Graphics and Interactive Media Research Group, ACM; May 3-7, 1992

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	InterTrust Solutions for A2b, InterTrust; Competitive Analysis AT&T/a2b music, Jun. 16, 1998; Email from Chris Drost-Hansen re press release: "AT&T Launches A2B Music Trial for Delivering Songs Over the Internet", Business Wire, Nov. 3, 1997; A2b's Recent Press Coverage, 1998
	Yes	ISO 11568-1 & -2: "Key management (retail) - Part 1: Introduction to key management"; and "- Part 2: Key management techniques for symmetric ciphers", ISO, Dec. 1, 1994
	Yes	ISO 13491-1: "Secure cryptographic devices (retail) – Part 1: Concepts, requirements and evaluation methods", ISO, Jun. 15, 1998
	Yes	ISO 8583-2: "Financial transaction card originated messages - Interchange message specifications - Part 2: Application and registration procedures for Institution Identification Codes (IIC)", ISO, Jul. 1, 1998
	Yes	ISO 8583-3: "Financial transaction card originated messages - Interchange message specifications - Part 3: Maintenance procedures for codes", ISO, Jul. 1, 1998
	Yes	ISO 9564-1 & -2: "Personal Identification Number (PIN) management and security - Part 1: Basic principals and requirements for online PIN handling in ATM and POS systems; & -2 Approved algorithm(s) for PIN encipherment", ISO, Apr. 15, 2002 & Dec. 15, 1991
	Yes	ISO 9807: "Banking and related financial services - Requirements for message authentication (retail)," ISO, Dec. 15, 1991
	Yes	Secure Electronic Transactions; Mastercard and Visa+C345
	Yes	Tanenbaum, Andrew S; van Renesse, Robbert; van Staveren, Hans; Sharp, Gregory J.; Mullender, Sape J.; Jansen, Jack; van Rossum, Guido; "Experiences with the Amoeba Distributed Operating System", Vrije Universiteit and Centrum voor Wiskunde en Informatica
	Yes	Tanenbaum, Andrew S; Mullender, Sape J.; van Renesse, Robbert; "Using Sparse Capabilities in a Distributed Operating System", Vrije Universiteit and Centre for Mathematics and Computer Science
Y	Yes	Tanenbaum, Andrew S; van Renesse, Robbert; van Staveren, Hans; Sharp, Gregory J.; Mullender, Sape J.; Jansen, Jack; van Rossum, Guido; "Amoeba System", Communications of the ACM, Vol. 33, No. 12, Dec. 1990
	Yes	"KeyKOS Principles of Operation", Key Logic document KL002-04, 1985, (Fourth Edition, Jan. 1987)
	Yes	Landau, Charles R.; "Security in a Secure Capability-Based System", Operating Systems Review, Oct. 1989
	Yes	"Security in KeyKOS"
	Yes	Hardy, Norman; "The Keykos Architecture", Key Logic Document KL028-08, Eighth Edition, Dec. 1990
	Yes	Johnson, Howard L.; Koegel, John F.; Koegel, Rhonda M; "A Secure Distributed Capability Based System", ACM, 1985

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Kim, Gene H.; Spafford, Eugene H.; "Experiences with Tripwire: Using Integrity Checkers for Intrusion Detection", COAST Laboratory, Purdue University, Feb. 22, 1995
	Yes	Blaze, Matt; "Key Management in an Encrypting File System", Proc. Summer 1994 USENIX Technical Conference, Jun. 1994
	Yes	Robinson, D.; Ullmann, R.; "Encoding Header Field for Internet Messages", Network Working Group RFC 1154, Apr. 1990; Rose, M.; McCloghrie, K.; "Structure and Identification of Management Information for TCP/IP-based Internets", Network Working Group RFC 1155, May 1990
	Yes	Rose, M.; McCloghrie, K.; "Structure and Identification of Management Information for TCP/IP-based Internets", Network Working Group RFC 1155, May 1990; McCloghrie, K.; Rose, M.; "Management Information Base for Network Management of TCP/IP-based internets", Network Working Group RFC 1156, May 1990; Case, J.; Fedor, M.; Schoffstall, M.; Davin, J.; "A Simple Network Management Protocol (SNMP)", Network Working Group RFC 1157, May 1990
	Yes	Davin, J.; Galvin, J.; McCloghrie, K.; "SNMP Administrative Model", Network Working Group RFC 1351, Jul., 1992; Galvin, J.; McCloghrie, K.; Davin, J.; "SNMP Security Protocols", Network Working Group RFC 1352, Jul., 1992; McCloghrie, K.; Davin, J.; Galvin, J.; "Definitions of Managed Objects for Administration of SNMP Parties", Network Working Group RFC 1353, Jul., 1992
	Yes	"PKCS #1: RSA Encryption Standard", RSA Laboratories Technical Note, Ver. 1.5, Revised Nov. 1, 1993
	Yes	"PKCS #3: Diffie-Hellman Key-Agreement Standard", RSA Laboratories Technical Note, Ver. 1.4, Revised Nov. 1, 1993
	Yes	"PKCS #6: Extended-Certificate Syntax Standard", RSA Laboratories Technical Note, Ver. 1.5, Revised Nov. 1, 1993
	Yes	"PKCS #9: Selected Attribute Types", RSA Laboratories Technical Note, Ver. 1.1, Revised Nov. 1, 1993
	Yes	Shneier, B.; "Description of new variable-length key, 64-bit block cipher (Blowfish)", Fast Software Encryption, Cambridge Security Workshop Proceedings, 1994
	Yes	Feistel, H.; "Cryptographic Coding for Data-Bank Privacy", IBM document RC 2827, Mar. 18, 1970
	Yes	ACORN/ CLEAR, 1996-1998
	Yes	Tuck, Bill; "Electronic Copyright Management Systems: Final Report of a Scoping Study for eLib", Jul., 1996

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
Y	Yes	"CopySmart (CSM) suit", European Information Technology for Information Science; COPYSMART - 20517: "CITED based multi-media IPR management on cost effective smart device", European Information Technology for Information Science, start date Dec. 1, 1995; Summaries of Projects (FP III/IV) - Part I: "ESPIRIT Project 20517 - COPYSMART CITED based multi-media IPR management on cost effective smart device", European Information Technology for Information Science, Oct. 1998
	Yes	"CREANET - Creative Rights European Agency NETwork - Project Profile" information society technologies, edited Feb. 18, 2000
	Yes	"iOpener System Description", National Semiconductor, 1993
	Yes	"iPower Technology" (National Semiconductor marketing brochure)
	Yes	"The Standards Business: Time for Change," European Commission DG111 Espirit Project 5th Consensus Forum, Nov. 3-4, 1998; "ESPIRIT Project 20676 - IMPRIMATUR - Intellectual Multimedia Property Rights Model and Terminology for Universal Reference", IMPRIMATUR Consortium, Oct. 1998; Electronic Reserve Copyright Management System (ERCOMS), International Institute for Electronic Library Research, website updated by Ramsden, Anne, Jul. 22, 1996; Achievements Archive, www.imprimatur.net/ web pages; impramatur news, iMPRIMATUR, Dec. 1998; Poslon, Chris: "The Imprimatur Project"
	Yes	JUKEBOX-Music Across Borders, LIB-JUKEBOX/4-1049
	Yes	"ESPRIT Project 24378 - MENHIR European Multimedia network of high quality image registration", Museums On Line, Feb. 1, 1997
	Yes	"ESPIRIT Project 22226 - MUSE - Developing standardized digital media management, signaling and encryption systems for the European music sector", International Federation of the Phonographic Industry, Oct. 1998
	Yes	"STARFISH State of the Art Dinancial Services for the inHabitants of isolated areas - Project Profile" information society technologies, time schedule Jan. 21, 2000 - Jun. 30, 2002

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	"TALISMAN - Tracing Authors' rights by Labelling Image Services and Monitoring Access Network," ACTS Project No. AC019, Doc Reference AC019-THO-RGS-FR-P-001-b1, Sep. 25, 1998; Simon, C.; Goray, E.; Vercken, G.; Delivet, B.; Delaigle, JF.; Boucqueau, JM.; "Digital Images protection managment in a broadcast framework: Overview/TALISMAN solution," Thomson-CSF, RTBF, ART3000, UCL; "TALISMAN: Tracing Authors' rights by labelling image services and monitoring access network," ACTS, Swiss Participation in European Research Programmes, Sep. 1, 1995 - Aug. 31, 1998
	Yes	"TELENET TELEtraining platform (on NETworks) - Project Profile" information society technologies, time schedule Mar. 6, 2000 - Mar. 30, 2000; "Deliverable D3: Specification of the Infrastructure And explanation of trust and confidence building solutions" Ver. 0.1, Telenet, Jul. 18, 2000; Email from Edmond Kouka to Jean-Francois Boisson re Affaire BC-CreaNet; Feb. 10, 2001; Email from Bogdan Lutkiewicz to Jean-Francois Boisson re TELENET TELEtraining platform - Bogdan Lutkiewicz, Poland, Gdansk; Mar. 4, 2001
Y	Yes	Boisson, Jean-Francois; "Management of Intellectual Property Rights in the Electronic Commerce: Textile Design Sales And Other Similar Initiatives," EURITIS
	Yes	U.S. Patent No. 5,251,294; Oct. 5, 1993
	Yes	S.H. Low, N.F. Maxemchuk, J.T. Bassil, & L. O'Gorman, Document Marking and Identification Using Both Line and Word Shifting, Infocom 95, 1994
	Yes	Caroni, Maxemchuck & O'Gorman, Electronic Marking and Identification Techniques to Discourage Document Copying, Proc. Infocom 94, 1994
	Yes	Wagner, Fingerprinting, IEEE Symp. On Info. and Privacy, Apr., 93
	Yes	H. Berghal, L Ogorman, "Protecting Ownership Rights Through Digital Watermarking", IEEE Computing v. 29, No.7, Jul., 1996,
	Yes	Chor, Fiat & Naor, Tracing Traitors, Crypto 94, p. 257, 1994
	Yes	David Chaum, "Security Without Identification: Transaction Systems to Make Big Brother Obsolete", Comm. Of the ACM, vol. 28, no. 10, Oct. 1985
	Yes	"Wallet Databases with Observers", David Chaum, Advances in Cryptology – Proceedings of Crypto '92 (pp. 89-105), 1992
Y	Yes	Sirbu, Marvin; Tygar, J.D.; "NetBill: An Internet Commerce System Optimized for Network Delivered Services", Carnegie Mellon University
	Yes	Ulrich Kohl, Jeffrey Lotspiech, Marc Kaplan, "Safeguarding Digital Library Contents and Users", IBM Research Division, D-Lib Magazine, Sept. 97
	Yes	Daniel Schutzer, A Need for a Common Infrastructure: Digital Libraries and Electronic Commerce, Apr. 1996
	Yes	Michael Lesk, Digital Libraries Meet Electronic Commerce: On-Screen Intellectual Property, Dec. 15, 98

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Lorcan Dempsey & Stuart L. Weibel; The Warwick Metadata Workshop: A Framework for the Deployment of Resource Description, Jul./ Aug. 96
	Yes	"AT&T Smart Cards Systems & Solutions", AT&T, 1993
Y	Yes	Gemplus; "MCOS: Multi Application Chip Operating System - Introduction", Gemplus Card International, 1990
	Yes	Guillou, Louis C.; "Smart Cards and Conditional Access", Springer-Verlag, 1988
	Yes	David L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", 1981
	Yes	Kent, S.; "U.S. Department of Defense Security Options for the Internet Protocol", Network Working Group RFC 1108, Nov. 1991
	Yes	Deering, S.E.; "Host Extensions for IP Multicasting", Network Working Group, RFC 1112, Aug. 1989
	Yes	Pethia, R.; Crocker, S.; Fraser, B.; "Guidelines for the Secure Operation of the Internet", Network Working Group RFC 1281, Nov., 1991
	Yes	Galvin, J.; McCloghrie, K.; "Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)", Network Working Group RFC 1446, Apr., 1993
	Yes	Eastlake III, D.; "Physical Link Security Type of Service", Network Working Group RFC 1455, May, 1993
	Yes	Kastenholz, F.; "The Definitions of Managed Objects for the Security Protocols of the Point-to-Point Protocol", Network Working Group RFC 1472, Jun. 1993
Y	Yes	Kohl, J., Neuman, C.; "The Kerberos Network Authentication Service (V5)", Network Working Group RFC 1510, Sep., 1993
	Yes	Eastlake III, D.; Crocker, S.; Schiller, J.; "Randomness Recommendations for Security", Network Working Group RFC 1750, Dec. 1994
	Yes	Haller, N.; "The S/KEY One-Time Password System", Network Working Group RFC 1760, Feb., 1995
	Yes	Atkinson, R.; "Security Architecture for the Internet Protocol", Network Working Group RFC 1825, Aug., 1995
	Yes	Crocker, S.; Freed, N.; Galvin, J.; Murphy, S.; "MIME Object Security Services", Network Working Group RFC 1848, Oct., 1995
	Yes	U.S. Patent No. 5,251,294; Oct. 5, 1993
	Yes	S.H. Low, N.F. Maxemchuk, J.T. Bassil, & L. O'Gorman, "Document Marking and Identification Using Both Line and Word Shifting," AT&T Bell Laboratories, Infocom 95, Jul. 29, 1994
	Yes	Brassil, J.; Low, S.; Maxemchuck, N.; O'Gorman L.; "Electronic Marking and Identification Techniques to Discourage Document Copying," AT&T Bell Laboratories, Proc. Infocom 94, 1994
	Yes	Wagner, Neal; "Fingerprinting," Drexel University, IEEE Symp. On Info. and Privacy, Apr., 1993
	Yes	Berghal, Hal; Ogorman, Lawrence; "Protecting Ownership Rights Through Digital Watermarking," IEEE Computing v. 29, no.7, pp. 101-103, Jul., 1996
	Yes	Chor, Benny; Fiat, Amos; Naor, Moni; "Tracing Traitors," Crypto 94, p. 257, 1994

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Chaum, David; "Security Without Identification: Transaction Systems to Make Big Brother Obsolete", Communications of the ACM, vol. 28, no. 10, Oct., 1985
	Yes	Chaum, David; Pederson, Torben Pryds; "Wallet Databases with Observers", CWI, Aarhus University, David Chaum, Advances in Cryptology -- Proceedings of Crypto '92, pp. 89-105, 1992
	Yes	Kohl, Ulrich; Lotspiech, Jeffrey; Kaplan, Marc; "Safeguarding Digital Library Contents and Users", IBM Research Division, D-Lib Magazine, Sept., 1997
	Yes	Schutzer, Daniel; "A Need for a Common Infrastructure: Digital Libraries and Electronic Commerce," Citibank, D-Lib Magazine, Apr., 1996
	Yes	Paepcke, Andreas; "Summary of Stanford's Digital Library Testbed and Status", Stanford University, D-Lib Magazine, Jul.-Aug., 1996
	Yes	Dempsey, Lorcan; Weibel, Stuart L.; "The Warwick Metadata Workshop: A Framework for the Deployment of Resource Description", University of Bath, OCLC Office of Research, D-Lib Magazine, Jul.-Aug., 1996
	Yes	"AT&T Smart Cards Systems & Solutions", AT&T, 1993
	Yes	Brad J. Cox, Dr., "What if there is a silver bullet?", Dobbs Journal, Oct. 1992
	Yes	Guillou, Louis C.; "Smart Cards and Conditional Access", Springer-Verlag, 1988
	Yes	Chaum, David; "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", Communications of the ACM, vol. 24, No. 3, Feb., 1981
	Yes	Kent, S.; "U.S. Department of Defense Security Options for the Internet Protocol", Network Working Group RFC 1108, Nov. 1991
	Yes	Deering, S.; "Host Extensions for IP Multicasting", Network Working Group RFC 1112, Aug. 1989
	Yes	White, Steve R.; Comerford, Liam; "ABYSS: A Trusted Architecture for Software Protection", IEEE, Apr. 27, 1987
	Yes	Ross, Philip E.; "Cops versus robbers in cyberspace"; Forbes, Sep. 9, 1996
	Yes	"Data Networks and Open System Communications, Directory: Information Technology -- Open Systems Interconnection -- The Directory: Overview of Concepts, Models, and Services", ITU-T Recommendation X.500, International Telecommunication Union, Nov. 1993
	Yes	Bender, W.; Gruhl, D.; Morimoto, N.; Lu, A.; "Techniques for data hiding", IBM Systems Journal, Vol. 35, Nos. 3&4, 1996
	Yes	Maxemchuk, N.F.; "Electronic Document Distribution", AT&T Bell Laboratories
	Yes	Doster, Bill; Rees, Jim; "Third-Party Authentication in the Institutional File System", Center for Information Technology Integration
	Yes	Levy, Steven; "E-Money (That's What I Want)", Wired Magazine, Issue 2.12, Dec. 94
	Yes	Arms, William Y., "Key Concepts in the Architecture of the Digital Library", D-Lib Magazine, Jul. 1995
	Yes	Weingart, S.H., "Physical Security for the uABYSS System", IEEE, 1987
	Yes	B. Strohm, L. Comerford, S. R. White, "ABYSS: Tokens", IBM Research Report Number RC 12402, Dec. 18, 1986

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Gozani, Shai; Gray, Mary; Keshav, Srinivasan; Madisetti, Mijay; Munson, Ethan; Rosenblum, Mendel; Schoettler, Steve; Sullivan, Mark; Terry, Douglas; "GAFFES: The Design of a Globally Distributed File System", Report No. UCB/CSD 87/361; Computer Science Division (EECS), U.C. Berkley, Jun. 1997
	Yes	Chaum, David; Fiat, Amos; Naor, Moni; "Untraceable Electronic Cash", Lecture Notes in Computer Science, 403, Advances in Cryptology - CRYPTO '88 Proceedings, 1988
	Yes	Chaum, David; "Privacy and Social Protection in Electronic Payment Systems", Chapter 12, The Future of Money in the Information Age
	Yes	Bos, Jurjen.; Chaum, David; "SmartCash: a Practical Electronic Payment System", Center for Mathematics and Computer Science, Report CS-R9035, Aug.
	Yes	Gircys, Gintaras R.; Understanding and Using COFF, O'Reilly & Associates, Inc.; Nov. 1988
	Yes	Unix System V, Release 3.2, Programmer's Guide Vol. II, AT&T, Prentice Hall, 1989
	Yes	Richarson, Dennis W.; Electric Money: Evolution of an Electronic Funds-Transfer System, The MIT Press, 1970
	Yes	Custer, Helen; Inside Windows NT, Microsoft Press, Redmond, WA, 1993
	Yes	Pietrek, Matt; Windows Internals: The Implementation of the Windows Operating Environment, Addison-Wesley, 1993
	Yes	Gilde, R., "DAT-Heads: Frequently Asked Questions", 1991, Release 3.1-Sep. 2, 1992
	Yes	Tardo, Joseph; Valente, Luis; "Mobile Agent Security and Telescript", General Magic, Inc.
	Yes	"Telescript Security", BYTE.com, Oct. 1994
	Yes	"Forum on Risks to the Public in Computers and Related Systems: ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator", Risks-Forum Digest, Vol. 15, Issue 40, Jan. 24, 1994
	Yes	Sahuguet, Arnaud; "Piracy: the Dark Side of Electronic Commerce: CIS-700/2", Univ. of Pennsylvania, May 5, 1998
Y	Yes	Rouaix, Francois; "A Web navigator with applets in Caml", INRIA
	Yes	Fuchsberger, Andreas; Gollmann, Dieter; Lothian, Paul; Paterson, Kenneth G.; Sidiropoulos, Abraham; "Public-key Cryptography on Smart Cards", Information Security Group
	Yes	"An Introduction to Safety and Security in Telescript", Telescript Powered
	Yes	Clarke, Roger; Bunting, Angela; "Cryptography issues in plain text", Privacy Law and Policy Reporter, 1996
Y	Yes	Pratt & Witney Inuse
Y	Yes	Use of ATM
Y	Yes	Use of Set Top Box
Y	Yes	Protective Envelope System
		PRIOR ART
	Yes	3,573,747; Adams et al.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	3,609,697; Blevins
	Yes	3,790,700; Callais et al.
	Yes	3,796,830; Smith
	Yes	3,798,359; Feistel
	Yes	3,798,360; Feistel
	Yes	3,798,605; Feistel
	Yes	3,806,882; Clarke
	Yes	3,829,833; Freeny, Jr.
	Yes	3,906,448; Henriques
	Yes	3,911,397; Freeny, Jr.
	Yes	3,924,065; Freeny, Jr.
	Yes	3,931,504; Jacoby
	Yes	3,946,200; Brobeck et al.
	Yes	3,946,220; Brobeck et al.
	Yes	3,956,615; Anderson et al.
	Yes	3,958,081; Ehrsam et al.
	Yes	3,970,992; Boothroyd et al.
	Yes	4,048,619; Forman, Jr. et al.
	Yes	4,071,911; Mazur
	Yes	4,112,421; Freeny, Jr.
	Yes	4,120,030; Johnstone
	Yes	4,162,483; Entenman
	Yes	4,163,280; Mori et al.
	Yes	4,168,396; Best
	Yes	4,196,310; Forman et al.
	Yes	4,200,913; Kuhar et al.
	Yes	4,209,787; Freeny, Jr.
	Yes	4,217,588; Freeny, Jr.
	Yes	4,220,991; Hamano et al.
	Yes	4,232,193; Gerard
	Yes	4,232,317; Freeny, Jr.
	Yes	4,236,217; Kennedy
	Yes	4,253,157; Kirschner et al.
	Yes	4,262,329; Bright et al.
	Yes	4,265,371; Desai et al.
	Yes	4,270,182; Asija
	Yes	4,278,837; Best
	Yes	4,305,131; Best
	Yes	4,306,289; Lumley
	Yes	4,309,569; Merkle
	Yes	4,319,079; Best
	Yes	4,323,921; Guillou
	Yes	4,328,544; Baldwin et al.
	Yes	4,337,483; Guillou
	Yes	4,361,877; Dyer et al.
	Yes	4,375,579; Davida et al.
	Yes	4,433,207; Best
	Yes	4,434,464; Suzuki et al.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	4,442,486; Mayer
	Yes	4,446,519; Thomas
	Yes	4,454,594; Heffron et al.
	Yes	4,458,315; Uchenick
	Yes	4,462,076; Smith, III
	Yes	4,462,078; Ross
	Yes	4,465,901; Best
	Yes	4,471,163; Donald et al.
	Yes	4,484,217; Block et al.
	Yes	4,494,156; Kadison et al.
	Yes	4,513,174; Herman
	Yes	4,528,588; Lofberg
	Yes	4,528,643; Freeny, Jr.
	Yes	4,553,252; Egendorf
	Yes	4,558,176; Arnold et al.
	Yes	4,558,413; Schmidt et al.
	Yes	4,562,306; Chou et al.
	Yes	4,562,495; Bond et al.
	Yes	4,577,289; Comerford et al.
	Yes	4,584,641; Guglielmino
	Yes	4,588,991; Atalla
	Yes	4,589,064; Chiba et al.
	Yes	4,593,183; Fukatsu
	Yes	4,593,353; Pickholtz
	Yes	4,593,376; Volk
	Yes	4,595,950; Lofberg
	Yes	4,597,058; Izumi et al.
	Yes	4,622,222; Johnson
	Yes	4,634,807; Chorley et al.
	Yes	4,644,493; Chandra et al.
	Yes	4,646,234; Tolman et al.
	Yes	4,652,990; Pailen et al.
	Yes	4,658,093; Hellman
	Yes	4,670,857; Rackman
	Yes	4,672,572; Alsberg
	Yes	4,677,434; Fascenda
	Yes	4,677,552; Sibley, Jr.
	Yes	4,680,731; Izumi et al.
	Yes	4,683,553; Mollier
	Yes	4,685,056; Barnsdale et al.
	Yes	4,688,169; Joshi
	Yes	4,691,350; Kleijne et al.
	Yes	4,696,034; Wiedemer
	Yes	4,700,296; Palmer, Jr. et al.
	Yes	4,701,846; Ikeda et al.
	Yes	4,712,238; Gilhousen et al.
	Yes	4,713,753; Boebert et al.
	Yes	4,727,550; Chang et al.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	4,740,890; William
	Yes	4,747,139; Taaffe
	Yes	4,757,533; Allen et al.
	Yes	4,757,534; Matyas et al.
	Yes	4,768,087; Taub et al.
	Yes	4,791,565; Dunham et al.
	Yes	4,796,181; Wiedemer
	Yes	4,798,209; Klingenbeck et al.
	Yes	4,799,156; Shavit et al.
	Yes	4,807,288; Ugon et al.
	Yes	4,817,140; Chandra et al.
	Yes	4,823,264; Deming
	Yes	4,827,508; Shear
	Yes	4,858,121; Barber et al.
	Yes	4,864,494; Kobus
	Yes	4,866,769; Karp
	Yes	4,868,877; Fischer
	Yes	4,903,296; Chandra et al.
	Yes	4,924,378; Hershey et al.
	Yes	4,930,073; Cina, Jr.
	Yes	4,949,187; Cohen
	Yes	4,975,647; Downer et al.
	Yes	4,977,594; Shear
	Yes	4,999,806; Chernow et al.
	Yes	5,001,752; Fischer
	Yes	5,005,122; Griffin et al.
	Yes	5,005,200; Fischer
	Yes	5,010,571; Katznelson
	Yes	5,023,907; Johnson et al.
	Yes	5,047,928; Wiedemer
	Yes	5,048,085; Abraham et al.
	Yes	5,050,213; Shear
	Yes	5,091,966; Bloomberg et al.
	Yes	5,103,392; Mori
	Yes	5,103,476; Waite et al.
	Yes	5,111,390; Ketcham
	Yes	5,119,493; Janis et al.
	Yes	5,126,936; Champion et al.
	Yes	5,128,525; Stearns et al.
	Yes	5,136,643; Fischer
	Yes	5,136,646; Haber et al.
	Yes	5,136,647; Haber et al.
	Yes	5,136,716; Harvey et al.
	Yes	5,146,575; Nolan, Jr.
	Yes	5,148,481; Abraham et al.
	Yes	5,155,680; Wiedemer
Y	Yes	5,163,091; Graziano et al.
	Yes	5,168,147; Bloomberg

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	5,185,717; Mori
	Yes	5,187,787; Skeen et al.
	Yes	5,201,046; Goldberg et al.
	Yes	5,201,047; Maki et al.
	Yes	5,208,748; Flores et al.
	Yes	5,214,702; Fischer
	Yes	5,216,603; Flores et al.
	Yes	5,221,833; Hecht
	Yes	5,222,134; Waite et al.
	Yes	5,224,160; Paulini et al.
	Yes	5,224,163; Gasser et al.
	Yes	5,227,797; Murphy
	Yes	5,235,642; Wobber et al.
	Yes	5,241,671; Reed et al.
	Yes	5,245,165; Zhang
	Yes	5,247,575; Sprague et al.
	Yes	5,257,369; Skeen et al.
	Yes	5,260,999; Wyman
	Yes	5,263,158; Janis
	Yes	5,265,164; Matyas et al.
Y	Yes	5,276,735; Boebert et al.
	Yes	5,280,479; Mary
	Yes	5,285,494; Sprecher et al.
	Yes	5,301,231; Abraham et al.
	Yes	5,311,591; Fischer
	Yes	5,319,705; Halter et al.
	Yes	5,319,785; Halter et al.
	Yes	5,335,169; Chong
	Yes	5,337,360; Fischer
	Yes	5,341,429; Stringer et al.
	Yes	5,343,527; Moore
	Yes	5,347,579; Blandford
	Yes	5,351,293; Michener et al.
Y	Yes	5,355,474; Thuraisngham et al.
	Yes	5,365,587; Campbell et al.
	Yes	5,373,440; Cohen et al.
	Yes	5,373,561; Haber et al.
	Yes	5,390,247; Fischer
	Yes	5,390,330; Talati
	Yes	5,392,220; van den Hamer et al.
	Yes	5,392,390; Crozier
	Yes	5,394,469; Nagel et al.
	Yes	5,410,598; Shear
	Yes	5,412,717; Fischer
	Yes	5,418,713; Allen
	Yes	5,420,927; Micali
	Yes	5,421,006; Jablon
	Yes	5,422,953; Fischer

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	5,428,606; Moskowitz
	Yes	5,438,508; Wyman
	Yes	5,442,645; Ugon
	Yes	5,444,779; Daniele
	Yes	5,449,895; Hecht et al.
	Yes	5,449,896; Hecht et al.
	Yes	5,450,493; Maher
	Yes	5,453,601; Rosen
	Yes	5,453,605; Hecht et al.
	Yes	5,455,407; Rosen
	Yes	5,455,861; Faucher et al.
	Yes	5,455,953; Russell
	Yes	5,457,746; Dolphin
	Yes	5,457,747; Drexler et al.
	Yes	5,458,494; Krohn et al.
	Yes	5,463,565; Cookson et al.
	Yes	5,473,687; Lipscomb et al.
	Yes	5,473,692; Davis
	Yes	5,479,509; Ugon
	Yes	5,485,622; Yamaki
	Yes	5,491,800; Goldsmith et al.
	Yes	5,497,479; Hornbuckle
	Yes	5,497,491; Mitchell et al.
	Yes	5,499,298; Narasimhalu et al.
	Yes	5,504,757; Cook et al.
	Yes	5,504,818; Okano
	Yes	5,504,837; Griffeth et al.
	Yes	5,508,913; Yamamoto et al.
	Yes	5,509,070; Schull
	Yes	5,513,261; Maher
	Yes	5,517,518; Rosen
	Yes	5,530,235; Stefik et al.
	Yes	5,530,752; Rubin
	Yes	5,533,123; Force et al.
	Yes	5,534,855; Shockley et al.
	Yes	5,534,975; Stefik et al.
	Yes	5,535,322; Hecht
	Yes	5,537,526; Anderson et al.
	Yes	5,539,735; Moskowitz
	Yes	5,539,828; Davis
	Yes	5,550,971; Brunner et al.
	Yes	5,553,282; Parrish et al.
	Yes	5,557,518; Rosen
	Yes	5,557,798; Skeen et al.
	Yes	5,563,946; Cooper et al.
	Yes	5,568,552; Davis
	Yes	5,572,673; Shurts
	Yes	5,592,549; Naget et al.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	5,606,609; Houser et al.
	Yes	5,613,004; Cooperman et al.
	Yes	5,621,797; Rosen
	Yes	5,629,770; Brassil et al.
	Yes	5,629,980; Stefik et al.
	Yes	5,633,932; Davis et al.
	Yes	5,634,012; Stefik et al.
	Yes	5,636,292; Rhoads
	Yes	5,638,443; Stefik et al.
	Yes	5,638,504; Scott et al.
	Yes	5,640,546; Gopinath et al.
	Yes	5,655,077; Jones et al.
	Yes	5,678,170; Grube et al.
	Yes	5,687,236; Moskowitz et al.
	Yes	5,689,587; Bender et al.
Y	Yes	5,692,047; McManis
	Yes	5,692,180; Lee
	Yes	5,710,834; Rhoads
	Yes	5,715,403; Stefik
	Yes	5,721,788; Powell et al.
	Yes	5,732,398; Tagawa
	Yes	5,740,549; Reilly et al.
	Yes	5,745,604; Rhoads
	Yes	5,748,763; Rhoads
	Yes	5,748,783; Rhoads
	Yes	5,748,960; Fischer
	Yes	5,754,849; Dyer et al.
	Yes	5,757,914; McManis
	Yes	5,758,152; LeTourneau
Y	Yes	5,765,152; Erickson
	Yes	5,768,426; Rhoads
	Yes	5,774,872; Golden et al.
	Yes	5,819,263; Bromley et al.
	Yes	5,842,173; Strum et al.
	Yes	BE 9 004 79
	Yes	DE 3 803 982
	Yes	DE 3 803 982 A1
	Yes	EP 0 084 441
	Yes	EP 0 084 441 A1
	Yes	EP 0 128 672
	Yes	EP 0 128 672 A1
	Yes	EP 0 135 422
	Yes	EP 0 135 422 A1
	Yes	EP 0 180 460
	Yes	EP 0 180 460 A1
	Yes	EP 0 370 146
	Yes	EP 0 370 146 A1
	Yes	EP 0 399 822 A2

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	EP 0 421 409
	Yes	EP 0 421 409 A2
	Yes	EP 0 456 386
	Yes	EP 0 456 386 A2
	Yes	EP 0 469 864
	Yes	EP 0 469 864 A2
	Yes	EP 0 469 864 A3
	Yes	EP 0 565 314
	Yes	EP 0 565 314 A2
	Yes	EP 0 593 305
	Yes	EP 0 593 305 A2
	Yes	EP 0 651 554
	Yes	EP 0 651 554 A1
	Yes	EP 0 668 695
	Yes	EP 0 668 695 A2
	Yes	EP 0 668 695 A3
	Yes	EP 0 695 985
	Yes	EP 0 695 985 A1
	Yes	EP 0 696 798
	Yes	EP 0 696 798 A1
	Yes	EP 0 714 204
	Yes	EP 0 714 204 A2
	Yes	EP 0 715 243
	Yes	EP 0 715 243 A1
	Yes	EP 0 715 244
	Yes	EP 0 715 244 A1
	Yes	EP 0 715 245
	Yes	EP 0 715 245 A1
	Yes	EP 0 715 246
	Yes	EP 0 715 246 A1
	Yes	EP 0 715 247
	Yes	EP 0 715 247 A1
	Yes	EP 0 725 376
	Yes	EP 0 725 376 A2
	Yes	EP 0 749 081
	Yes	EP 0 749 081 A1
	Yes	EP 0 763 936
	Yes	EP 0 763 936 A2
	Yes	EP 0 778 513
	Yes	EP 0 778 513 A2
	Yes	EP 0 795 873
	Yes	EP 0 795 873 A2
	Yes	EP 0 800 312
	Yes	EP 0 800 312 A1
	Yes	GB 2,136,175
	Yes	GB 2,264,796
	Yes	GB 2,294,348
	Yes	GB 2,295,947

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	JP 01-068835
	Yes	JP 02-242352
	Yes	JP 02-247763
	Yes	JP 02-294855
	Yes	JP 04-369068
	Yes	JP 05-181734
	Yes	JP 05-257783
	Yes	JP 05-268415
	Yes	JP 06-175794
	Yes	JP 06-215010
	Yes	JP 06-225059
	Yes	JP 07-056794
	Yes	JP 07-084852
	Yes	JP 07-141138
	Yes	JP 07-200317
	Yes	JP 07-200492
	Yes	JP 07-244639
	Yes	JP 08-137795
	Yes	JP 08-152990
	Yes	JP 08-185292
	Yes	JP 08-185298
	Yes	JP 57-726
	Yes	JP 62-241061
	Yes	WO 85/02310
	Yes	WO 85/03584
	Yes	WO 90/02382
	Yes	WO 92/06438
	Yes	WO 92/22870
	Yes	WO 93/01550
	Yes	WO 94/01821
	Yes	WO 94/03859
	Yes	WO 94/06103
	Yes	WO 94/16395
	Yes	WO 94/18620
	Yes	WO 94/22266
	Yes	WO 94/27406
	Yes	WO 95/14289
	Yes	WO 96/00963
	Yes	WO 96/03835
	Yes	WO 96/05698
	Yes	WO 96/06503
	Yes	WO 96/13013
	Yes	WO 96/21192
	Yes	WO 96/24092
	Yes	WO 97/03423
	Yes	WO 97/07656
	Yes	WO 97/25816
	Yes	WO 97/32251

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	WO 97/48203
	Yes	Amerke, David, et al., News Release, AT&T, Jan. 9, 1995, AT&T encryption system protects information services, 1 page.
	Yes	Applications Requirements for Innovative Video Programming; How to Foster (or Cripple) Program Development Opportunities for Interactive Video Programs Delivered on Optical Media; A Challenge for the Introduction of DVD (Digital Video Disc) (19-20 Oct. 19
	Yes	Argent Information Q&A Sheet, http://www.digital-watermark.com/, Copyright 1995, The DICE Company, 7 pages.
	Yes	Automation of Securities Markets and Regulatory Implications, Financial Market Trends, n50, p. 20-33, Oct. 1991. [File 148, Gale Group Trade & Industry DB, Dialog(R) commercial database]
	Yes	Avery et al, Recommender Systems For Evaluating Computer Messages, Communications of the ACM, pp. 88-89 (Mar. 1997).
	Yes	Background on the Administration's Telecommunications Policy Reform Initiative, News Release, The White House, Office of the President, Jan. 11, 1994
	Yes	Baggett, Claude, Cable's Emerging Role in the Information Superhighway, Cable Labs, 13 slides.
	Yes	Balabanovic et al, Content-based, Collaborative Recommendation, Communications of the ACM, pp. 66-72 (Mar. 1997).
	Yes	Barassi, Theodore Sedgwick Esq., The Cybernotary: Public Key Registration and Certificaiton and Authentication of International Legal Transactions, 4 pages.
	Yes	Barnes, Hugh, memo to Henry LaMuth, subject: George Gilder articles, May 31, 1994.
	Yes	Bart, Dan, Comments in the Matter of Public Hearing and Request for Comments on the International Aspects of the National Information Infrastructure, Before the Department of Commerce, Aug. 12, 1994.
	Yes	Baum, Michael, Worldwide Electronic Commerce: Law, Policy and Controls Conference, program details, Nov. 11, 1993.
	Yes	Best, Robert M., Digest of Papers, VLSI: New Architectural Horizons, Feb. 1980, Preventing Software Piracy With Crypto-Microprocessors, pp. 466-469.
	Yes	Bisbey, Richard L., II and Gerald J Popek, Encapsulation: An Approach to Operating System Security, (USC/Information Science Institute, Marina Del Rey, CA) Oct. 1973, pp. 666-675.
	Yes	Blom et al., Encryption Methods in Data Networks, Ericsson Technics, No. 2, 1978, Stockholm, Sweden.
	Yes	Bruner, Rick E., "PowerAgent, NetBot help advertisers reach Internet shoppers," Aug. 1997 (Document from Internet).
	Yes	Cable Television and America's Telecommunications Infrastructure, (National Cable Television Association, Washington, D.C.), Apr. 1993, 19 pages.
	Yes	Caruso, Denise, Technology, Digital Commerce: 2 plans for watermarks, which can bind proof of authorship to electronic works, N.Y. Times, Aug. 7, 1995, p. D5.
	Yes	CD ROM, Introducing . . . The Workflow CD-ROM Sampler, Creative Networks, MCIMail: Creative Networks, Inc., Palo Alto, California.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	CGI Common Gateway Interface Document from the Internet, <cgi@ncsa.uiuc.edu>, 1996, 1 page.
	Yes	Chase, Chevy, M.D., DiscStore (Electronic Publishing Resources 1991).
Y	Yes	Choudhury, et al., "Copyright Protection for Electronic Publishing over Computer Networks," AT&T Bell Laboratores, Murray Hill, New Jersey 07974 (Jun. 1994).
	Yes	Clark, Tim, Ad service gives cash back, Document from the Internet: <www.news.com./News/Item/0,4,13050,00.html> (visited Aug. 4, 1997), 2 pages.
	Yes	Codercard, Spec Sheet--Basic Coder Subsystem (Interstate Electronics Corp., Anaheim, CA), (undated) 4 pages.
	Yes	Collection of documents including: Protecting Electronically Published Properties, Increasing Publishing Profits, (Electronic Publishing Resources Inc.) Jan. 1993, 25 pages.
	Yes	Communications of the ACM, Intelligent Agents, Jul. 1994, vol. 37, No. 7.
	Yes	Communications of the ACM, Jun. 1996, vol. 39, No. 6.
	Yes	Computer Systems Policy Project (CSSP), Perpsectives on the National Information Infrastructure: Ensuring Interoperability (Feb. 1994), Feb. 1994.
	Yes	Cunningham, Donna, et al., News Release, AT&T, Jan. 31, 1995, AT&T, VLSI Technology join to improve info highway security, 3 pages.
	Yes	Data Sheet, About the Digital Notary Service, Surety Technologies, Inc., 1994-1995, 6 pages.
	Yes	Dempsey, et al., "The Warwick Metadata Workshop: A Framework for the Deployment of Resource Description", D-Lib Magazine, Jul. 15, 1996.
	Yes	Denning et al., Data Security, 11 Computing Surveys No. 3, Sep. 1979, pp. 227-249.
	Yes	Diffie, Whitfield and Martin E. Hellman, IEEE Transactions on Information Theory, vol. 22, No. 6, Nov. 1976, New Directions in Cryptography, pp. 644-651.
	Yes	Diffie, Whitfield and Martin E. Hellman, Proceedings of the IEEE, vol. 67, No. 3, Mar. 1979, Privacy and Authentication: An Introduction to Cryptography, pp. 397-427.
	Yes	DSP56000/DSP56001 Digital Signal Processor User's Manual, Motorola, 1990, pp. 2-2.
	Yes	Dusse, Stephen R. and Burton S. Kaliski, A Cryptographic Library for the Motorola 56000 in Damgard, I. M., Advances in Cryptology-Proceedings Eurocrypt 90, Springer-Verlag, 1991, pp. 230-244.
	Yes	Dyson, Esther, Intellectual Value, Wired Magazine, Jul. 1995, pp. 136-141 and 182-184.
	Yes	EDS Provides PowerAgent with Internet Services to Support One-to-One Marketing (PowerAgent Inc. 1997, no later than Aug. 13, 1997).
	Yes	EFFector Online vol. 6 No. 6, "A Publication of the Electronic Frontier Foundation," 8 pages, Dec. 6, 1993.
	Yes	EIA and TIA White Paper on National Information Infrastructure, published by the Electronic Industries Association and the Telecommunications Industry Association, Washington, D.C., no date.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Electronic Currency Requirements, XIWT (Cross Industry Working Group), (no date).
	Yes	Electronic Publishing Resources Inc. Protecting Electronically Published Properties Increasing Publishing Profits (Electronic Publishing Resources 1991).
	Yes	Firefly Network, Inc., www.ffly.com, What is Firefly? Firefly revision: 41.4 Copyright 1995, 1996.
	Yes	First CII Honeywell Bull International Symposium on Computer Security and Confidentiality, Jan. 26-28, 1981, Conference Text, pp. 1-21.
	Yes	Framework for National Information Infrastructure Services, Draft, U.S. Department of Commerce, Jul. 1994.
	Yes	Framework for National Information Infrastructure Services, NIST, Jul. 1994, 12 slides.
	Yes	Garcia, D. Linda, Science, space and technology, Hearing before Subcomm. on Technology, Environment, and Aviation, May 26, 1994 (testimony of D. Linda Garcia).
	Yes	Gleick, James, Dead as a Dollar, The New York Times Magazine, Jun. 16, 1996, Section 6, pp. 26-30, 35, 42, 50, 54.
	Yes	Greguras, Fred, Softic Symposium '95, Copyright Clearances and Moral Rights, Nov. 30, 1995 (as updated Dec. 11, 1995), 3 pages.
	Yes	Guillou, Louis C., Smart Cards and Conditional Access, Advances in Cryptography--Proceedings of EuroCrypt 84 (T. Beth et al, Ed., Springer-Verlag, 1985) pp. 480-490.
	Yes	Haar, Steven Vonder, PowerAgent Launches Commercial Service, Interactive Week Aug. 4, 1997, (Document from the Internet) 1 page.
	Yes	Harman, Harry H., Modern Factor Analysis, Third Edition Revised, University of Chicago Press, Chicago and London, 1976.
	Yes	Hearst, Interfaces For Searching the Web Scientific American pp. 68-72 (Mar. 1997).
	Yes	Herzberg, Amir et al., Public Protection of Software, ACM Transactions on Computer Systems, vol. 5, No. 4, Nov. 1987, pp. 371-393.
	Yes	Hofmann, Jud, Interfacing the NII to User Homes, (Consumer Electronic Bus. Committee) NIST, Jul. 1994, 12 slides.
	Yes	Hofmann, Jud, Interfacing the NII to User Homes, Electronic Industries Association, Consumer Electronic Bus Committee, 14 slides, no date.
	Yes	Holt, Stannie, Start-up promises user confidentiality in Web marketing service, Info World Electric, Aug. 13, 1997 (Document from Internet)/ (Infoworld Publishing Co. Aug. 4, 1997).
	Yes	HotJava.TM.: The Security Story Document from the Internet, (no date) 4 pages.
	Yes	How Can I Put an Access Counter on My Home Page?, World Wide Web FAQ, 1996, 1 page.
	Yes	Multimedia Mixed Objects Envelopes Supporting a Graduated Fee Scheme Via Encryption, IBM Technical Disclosure Bulletin, vol. 37, No. 3, Mar. 1, 1994, pp. 413-417, XP000441522.
	Yes	Transformer Rules Strategy for Software Distribution Mechanism-Support Products, IBM Technical Disclosure Bulletin, vol. 37, No. 48, Apr. 1994, pp. 523-525, XP000451335.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	IISP Break Out Session Report for Group No. 3, Standards Development and Tracking System, no date.
	Yes	Information Infrastructure Standards Panel: NII "The Information Superhighway", NationsBank--HGDeal--ASC X9, (no date), 15 pages.
	Yes	Intellectual Property and the National Information Infrastructure, a Preliminary Draft of the Report of the Working Group on Intellectual Property Rights, Green paper, Jul. 1994, 141 pages.
	Yes	Invoice? What's an Invoice?, Business Week, Jun. 10, 1996, pp. 110-112.
	Yes	Is Advertising Really Dead?, Wired 1.02, Part 2, 1994.
	Yes	Javasoft, Frequently Asked Questions--Applet Security, What's Java.TM.? Products and Services, Java/Soft News, Developer's Cornier, Jun. 7, 1996, 8 pages, Document from Internet, <java@java.sun.com.>
	Yes	Jiang, et al, A concept-Based Approach to Retrieval from an Electronic Industrial Directory, International Journal of Electronic Commerce, vol. 1, No. 1, Fall 1996, pp. 51-72.
	Yes	Jones, Debra, Top Tech Stories, PowerAgent Introducts First Internet `Infomediary` to Empower and Protect Consumers, Aug. 13, 1997, 3 pages (Document from Internet).
	Yes	Kautz, Referral Web: Combining Social Networks and Collaborative Filtering, Communications of the ACM, pp. 63-65 (Mar. 1997).
	Yes	Kelly, Kevin, Whole Earth Review, E-Money, pp. 40-59, Summer 1993.
	Yes	Kent, Stephen Thomas, Protecting Externally Supplied Software in Small Computers, (MIT/LCS/TR-255) Sep. 1980, 254 pages.
	Yes	Kohntopp, M., Sag's durch die Blume, Apr. 1996, marit@schulung.netuse.de
	Yes	Konstan et al, Applying Collaborative Filtering to Usenet News, Communications of the ACM, pp. 77-87 (Mar. 1997).
	Yes	Kristol et al., Anonymous Internet Mercantile Protocol, AT&T Bell Laboratories, Murray Hill, New Jersey, Draft: Mar. 17, 1994.
	Yes	Lagoze, Carl, D-Lib Magazine, Jul./Aug. 1996, The Warwick Framework, A Container Architecture for Diverse Sets of Metadata.
	Yes	Lanza, Mike, electronic mail, George Gilder's Fifth Article--Digital Darkhorse--Newspapers, Feb. 21, 1994.
	Yes	Levy, Steven, E-Money, That's What I want, WIRED, Dec. 1994, 10 pages.
	Yes	Low et al., Anonymous Credit Cards and its Collusion Analysis, AT&T Bell Laboratories, Murray Hill, New Jersey, Oct. 10, 1994.
	Yes	Low et al., Anonymous Credit Cards, AT&T Bell Laboratories, Proceedings of the 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, Nov. 2-4, 1994.
	Yes	Low et al., Document Marking and Identification using both Line and Word Shifting, AT&T Bell Laboratories, Murray Hill, New Jersey, Jul. 29, 1994.
	Yes	Lynch, Searching the Internet Scientific American pp. 52-56 (Mar. 1997).
	Yes	Maclachlan, Malcolm, PowerAgent Debuts Spam-Free Marketing, TechWire, Aug. 13, 1997, 3 pages (Document from Internet).
	Yes	Maxemchuk, Electronic Document Distribution, AT&T Bell Laboratories, Murray Hill, New Jersey 07974.
	Yes	Micro Card (Micro Card Technologies, Inc., Dallas, TX), (no date), 4 pages.
	Yes	Milbrandt, Eric, Stenanography Info and Archive, 1996, 2 pages.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
Y	Yes	Mori, Ryoichi and Masaji Kawahara, Superdistribution: The Concept and the Architecture, The Transactions of the EIEICI, V, E73, No. 7, Tokyo, Japan, Jul. 1990.
	Yes	Mossberg, Walter S., Personal Technology, Threats to Privacy On-Line Become More Worrisome, Wall Street Journal, Oct. 24, 1996.
	Yes	Negroponte, Nicholas, Electronic Word of Mouth, Wired, Oct. 1996, p. 218.
	Yes	Negroponte, Nicholas, Some Thoughts on Likely and Expected Communications Scenarios: A Rebuttal, Telecommunications, Jan. 1993, pp. 41-42.
	Yes	Neumann, et al., A Provably Secure Operating System: The System, Its Applications, and Proofs, Computer Science Labortory Report CSL-116, Second Edition, SRI International (May 1980).
	Yes	New Products, Systems and Services, AT&T Technology, vol. 9, No. 4, (undated), pp. 16-19.
	Yes	News from The Document Company Xerox, Xerox Announces Software Kit for Creating Working Documents with Dataglyphs Document from Internet, Nov. 6, 1995, 13 pages.
	Yes	NII, Architecture Requirements, XIWT, (no date).
	Yes	Open System Environment Architectural Framework for National Information Infrastructure Services and Standards, in Support of National Class Distributed Systems, Distributed System Engineering Program Sponsor Group, Draft 1.0, Aug. 5, 1994.
	Yes	Pelton, Dr. Joseph N., Telecommunications, Why Nicholas Negroponte is Wrong About the Future of Telecommunication, pp. 35-40, Jan. 1993.
	Yes	Portland Software's ZipLock, Internet Information, Copyright Portland Software, 1996-1997, 12 pages.
	Yes	PowerAgent Introduces First Internet `Infomediary` to Empower and Protect Consumers (PowerAgent Inc. Aug. 4, 1997).
	Yes	PowerAgent Introduces First Internet `Infomediary` to Empower and Protect Consumers (PowerAgent Inc., 1997 (no later than Aug. 13, 1997).
	Yes	PowerAgent Introduces First Internet `Infomediary` to Empower and Protect Consumers (Tech Talk Aug. 4, 1997).
	Yes	PowerAgent Introduces First Internet `Infomediary` to Empower and Protect Consumers (Techmall.com, Aug. 4, 1997).
	Yes	PowerAgent Introduces Internet's First True 1:1 Marketing Network (PowerAgent Inc., Aug. 4, 1997).
	Yes	PowerAgent Press Releases, "What the Experts are Reporting on PowerAgent," Aug. 13, 1997, 3 pages (Document from Internet).
	Yes	PowerAgent Press Releases, What the Experts are Reporting on PowerAgent, Aug. 13, 1997, 6 pages (Document from Internet).
	Yes	PowerAgent Press Releases, What the Experts are Reporting on PowerAgent, Aug. 4, 1997, 5 pages (Document from Internet).
	Yes	Premenos Announces Templar 2.0--Next Generation Software for Secure Internet EDI, Document from Internet: <webmaster@templar.net>, Jan. 17, 1996, 1 page.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Premenos Corp. White Paper: The Future of Electronic Commerce, A Supplement to Midrange Systems, Document from Internet, <webmaster@premenos.com>, 4 pages, no date.
	Yes	Press Release, "National Semiconductor and EPR Partner For Information Metering/Data Security Cards" (Mar. 4, 1994).
	Yes	Proper Use of Consumer Information on the Internet, Document from the Internet, White Paper, (PowerAgent Inc., Melo Park, CA) Jun. 1997, 9 pages.
	Yes	Rankine, Gordon, "Thomas--A Complete Single-Chip RSA Device," Advances in Cryptography, Proceedings of Crypto 86, pp. 480-487 (A.M. Odlyzko Ed., Springer-Verlag 1987).
	Yes	Reilly, Arthur K., Standards committee T1-Telecommunications, Input to the `International Telecommunications Hearings,` Panel 1: Component Technologies of the NII/GII, no date.
	Yes	Resnick, et al., Recommender Systems,Communications of the ACM, vol. 40, No. 3, Mar. 1997, pp. 56-89.
	Yes	Resnick, Filtering the Information On the Internet Scientific American pp. 62-64 (Mar. 1997).
	Yes	ROI-Solving Critical Electronic Publishing Problems (Personal Library Software, 1987 or 1988).
	Yes	Rose, Lance, Cyberspace and the Legal Matrix: Laws or Confusion?, 1991.
	Yes	Rosenthal, Steve, Interactive Newtork: Viewers Get Involved, New Media, Dec. 1992, pp. 30-31.
	Yes	Rosenthal, Steve, Interactive TV: The Gold Rush is on, New Media, Dec. 1992, pp. 27-29.
	Yes	Rosenthal, Steve, Mega Channels, New Media, Sep. 1993, pp. 36-46.
	Yes	Rothstein, Edward, Technology, Connections, Making the Internet come to you through `push` technology, New York Times, Jan. 20, 1997, p. D5.
	Yes	Rucker et al, Personalized Navigation For the Web, Communications of the ACM, pp. 73-75 (Mar. 1997).
	Yes	Rutkowski, Ken, PowerAgent Introduces First Internet `Infomediary` to Empower and Protect Consumers, Tech Talk News Story, Aug. 4, 1997, 1 page. (Document from Internet)
	Yes	Sager, Ira (Edited by), Bits & Bytes, Business Week, Sep. 23, 1996, p. 142E.
	Yes	Schlosstein, Steven, America: The G7's Comeback Kid, International Economy, Jun./Jul. 1993, 5 pages.
	Yes	Schurmann, Jurgen, Pattern Classification, A Unified View of Statistical and Neural Approaches, John Wiley & Sons, Inc., 1996.
	Yes	Scnaumueller-Bichl, Ingrid, et al., A Method of Software Protection Based on the Use of Smart Cards and Cryptographic Techniques, (undated), 9 pages.
	Yes	Serving the Community: A Public-Interest Vision of the National Information Infrastructure, Computer Professionals for Social Responsibility, Executive Summary, no date.
	Yes	Shear, Victor, Solutions for CD-ROM Pricing and Data Security Problems, pp. 530-533, CD ROM Yearbook 1988-1989) (Microsoft Press 1988 or 1989).

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Siuda, Karl, Security Services in Telecommunications Networks, Seminar: Mapping New Applications Onto New Technologies, edited by B. Plattner and P Gunzburger; Zurich, Mar. 8-10, 1988, pp. 45-52, XP000215989.
	Yes	Smith, Sean and J.D. Tyger, Signed Vector Timestamps: A Secure Protocol for Partial Order Time, CMU-93-116, School of Computer Science Carnegie Mellon University, Pittsburgh, Pennsylvania, Oct. 1991; version of Feb. 1993, 15 pages.
	Yes	Special Report, "The Internet: Fulfilling the Promise"; Lynch, Clifford; "The Internet Bringing Order From Chaos"; Resnick, Paul; "Search the Internet", Hearst, Marti A; "Filtering Information on the Internet"; Stefik, Mark; "Interfaces for Searching the
	Yes	Stefik, Mark, Introduction to Knowledge Systems, Chapter 7, Classification (Morgan Kaufmann Publishers, Inc., 1995) pp. 543-607.
	Yes	Stefik, Mark, Letting Loose the Light: Igniting Commerce in Electronic Publication, (Xerox PARC, Palo Alto, CA) 1994-1995, 35 pages.
	Yes	Stefik, Mark, Letting Loose the Light: Igniting Commerce In Electronic Publication, Internet Dreams: Archetypes, Myths, and Metaphors. Massachusetts Institute of Technology, 1996, pp. 219-253.
	Yes	Stefik, Trusted Systems Scientific American pp. 78-81 (Mar. 1997).
	Yes	Stephenson, Tom, Advanced Imaging, The Info Infrastructure Initiative: Data SuperHighways and You, pp. 73-74, May 1993.
	Yes	Sterling, Bruce, "Literary freeware: Not for Commercial Use", remarks at Computers, Freedom and Privacy Conference IV, Chicago, Mar. 26, 1994.
	Yes	Struif, Bruno, The Use of Chipcards for Electronic Signatures and Encryption, Proceedings for the 1989 Conference on VLSI and Computer Peripherals, IEEE Computer Society Press, 1989, pp. (4)155-(4)158.
	Yes	Templar Software and Services: Secure, Reliable, Standards-Based EDI Over the Internet, Prementos, Internet info@templar.net,.1page.
	Yes	Templar Overview,: Premenos, Internet info@templar.net, 4 pages.
	Yes	Terveen et al, A System For Sharing Recommendations, Communications of the ACM, pp. 59-62 (Mar. 1997).
	Yes	The 1:1 Future of the Electronic Marketplace: Return to a Hunting and Gathering Society, 2 pages, no date.
	Yes	The Benefits of ROI For Database Protection and Usage Based Billing (Personal Library Software, 1987 or 1988).
	Yes	The New Alexandria No. 1, Alexandria Institute, pp. 1-12, Jul.-Aug. 1986.
	Yes	This Web Agent Knows What You Like, Business Week, p. 142E (Sep. 23, 1996).
	Yes	Tygar, J.D. and Bennet Yee, Cryptography: It's Not Just For Electronic Mail Anymore, CMU-CS-93-107, School of Computer Science Carnegie Mellon University, Pittsburgh, PA, Mar. 1, 1993, 21 pages.
	Yes	Tygar, J.D. and Bennet Yee, Dyad: A System for Using Physically Secure Coprocessors, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (undated), 41 pages.
	Yes	Tygar, J.D. and Bennet Yee, Dyad: A System for Using Physically Secure Coprocessors, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 1991, 36 pages.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

InterTrust Tech. Corp. v. Microsoft Corp.
Case No. C 01-1640 SBA (MEJ)

APPENDIX OF PRIOR ART*

Anticipates	Renders Obvious	Description
	Yes	Valovic, T., The Role of Computer Networking in the Emerging Virtual Marketplace, Telecommunications, (undated), pp. 40-44.
	Yes	Voight, Joan, Beyond the Banner, Wired, Dec. 1996, pp. 196, 200, 204.
Y	Yes	Weber, Metering Technologies for Digital Intellectual Property, A Report to the International Federation of Reproduction Rights Organisations, pp 1-29; Oct. 1994, Boston, MA, USA.
Y	Yes	Weber, Robert, Digital Rights Management Technologies, A Report to the International Federation of Reproduction Rights Organisations, Northeast Consulting Resources, Inc., Oct. 1995, 49 pages.
	Yes	Weber, Robert, Document from the Internet: Digital Rights Management Technologies, Oct. 1995, 21 pages.
	Yes	Weder, Adele, Life on the Infohighway, INSITE, (no date), pp. 23-25.
	Yes	Weingart, Steve H., Physical Security for the ABYSS System, (IBM Thomas J. Watson Research Center, Yorktown Heights, NY), 1987, pp. 52-58.
	Yes	Weitzner, Daniel J., A Statement on EFF's Open Platform Campaign as a Nov., 1993, 3 pages.
	Yes	WEPIN Store, Stenography (Hidden Writing), Document from Internet: (Common Law), 1995, 1 page.
	Yes	White, Steve R., ABYSS: A Trusted Architecture for Software Protection, (IBM Thomas J. Watson Research Center, Yorktown Heights, NY), 1987, pp. 38-50.
	Yes	XIWT Cross Industry Working Team, 5 pages, Jul. 1994.
	Yes	Yee, Bennet, Using Secure Coprocessors, CMU-CS-94-149, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1994, 94 pages.
	Yes	Yellin, Frank, Document from the Internet: Low Level Security in Java, Sun Microsystems, 1996, 8 pages.

* Any possible "Y"s that were missed shall not negate the anticipatory nature of a reference, particularly where there is a chart in Appendix B.

EXHIBIT B to "DEFENDANT MICROSOFT CORPORATION's PRELIMINARY INVALIDITY CONTENTIONS (Patent Local Rules 3-3 and 3-4)" is provided electronically, via CD-ROM submitted herewith.

EXHIBIT C to "DEFENDANT MICROSOFT CORPORATION's PRELIMINARY INVALIDITY CONTENTIONS (Patent Local Rules 3-3 and 3-4)" is provided electronically, via CD-ROM submitted herewith.